MASTER EN SCIENCES INFORMATIQUES

Méthodologie pour la supervision de systèmes d'information

Théate, Marc

Award date:
2001

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX,

NAMUR

INSTITUT D'INFORMATIQUE

RUE GRANDGAGNAGE, 21, B-5000 NAMUR (BELGIUM)

Méthodologie pour la supervision de systèmes d'information

Marc THEATE

Mémoire présenté en vue de l'obtention du grade de

Licencié en Informatique

Année Académique 2000 - 2001

Résumé

Ce document traite d'une méthodologie destinée à mener de bout en bout des projets informatiques de supervision de systèmes d'information. Cette méthodologie vise à répondre aux besoins du département informatique ESM, Enterprise System Management, de la Fortis Banque en charge de tels projets mais peut aisément être généralisée à d'autres contextes organisationnels.

La première partie de ce document décrit le cadre d'étude de ce travail, des notions de supervision et fixe les objectifs de la méthodologie. La seconde partie va s'attacher à étudier des méthodologies existantes. Les troisième et quatrième parties décrivent les postulats ainsi que la méthodologie proposée. Enfin, la cinquième partie étudie le comportement de la méthodologie dans trois cas typiques de projet de supervision.

Abstract

This document deals with a methodology intended to take one through monitoring IT projects for information systems from start to finish. This methodology seeks to target the needs of ESM, Enterprise System Management, the IT department of Fortis Bank in charge of such projects, but can also be widely used in other organisational situations.

The first part of this document describes the framework of this work, fundamentals of monitoring and establishes the methodology objectives. The second part is devoted to a study of existing methodologies. The third and fourth parts outline the assumptions as well as the proposed methodology. Finally, the fifth part makes a study of the methodology performance in three typical project monitoring cases.

Avant-propos

Je tiens à remercier,

Monsieur Lesuisse, promoteur de ce mémoire, pour sa patience, ses conseils et sa supervision.

Madame d'Udekhem-Gevers et Monsieur Leclercq pour leurs conseils lors du séminaire de préparation aux mémoires.

Messieurs Marc Chalmagne, responsable de l'entité *IS Central Systems* et Guy Liégeois, chef du service *Enterprise System Management* chez Fortis Banque, pour m'avoir permis de réaliser ce mémoire dans les meilleures conditions et pour leur soutien durant les différentes étapes de ce projet.

Madame Marie Collard, responsable de la cellule *Availability*, qui m'a consacré une partie de son temps pour me permettre de cerner au mieux les différents problèmes auxquels elle est confrontée.

Les membres du jury, pour l'intérêt qu'ils porteront à ce travail.

Et tout spécialement

Mon épouse pour son soutien constant tout au long de ces années d'études.

Table des matières

Table des figures

Introduction

Dans ce document, nous allons traiter d'une méthodologie destinée à mener de bout en bout des projets informatiques de supervision de systèmes d'information. Elle est réalisée afin de répondre aux besoins du département informatique ESM, Enterprise System Management, de la Fortis Banque en charge de tels projets.

Cette étude s'articule autour de cinq grandes parties. La première partie décrit le cadre d'étude de ce travail. Elle présente l'entreprise, son département informatique et, en particulier, le département ESM et sa cellule *Availability*. De par son lien étroit avec le monde de la supervision, nous allons également décrire l'organisation et le fonctionnement du support aux utilisateurs. Un second chapitre sera entièrement consacré à l'explication des notions de supervision afin d'introduire le vocabulaire nécessaire à la bonne compréhension de ce travail, d'expliquer en quoi consiste la supervision de systèmes d'information et de présenter les outils de supervision *Tivoli* utilisés par le département ESM. Enfin, le dernier chapitre de cette partie va s'atteler à analyser le mode de fonctionnement et les processus de développement de la cellule *Availability*. Cet audit va permettre de mettre en évidence tous les problèmes que rencontre la cellule afin de pouvoir dégager les objectifs que devra atteindre la méthodologie.

Avant de construire une méthodologie de toute pièce, il nous a paru judicieux de voir si une telle méthodologie n'existait déjà pas sur le marché informatique. La deuxième partie de ce travail va donc s'atteler à étudier quatre méthodologies et à analyser dans quelle mesure celles-ci sont susceptibles d'apporter des réponses ou parties de réponse aux problèmes que connaît la cellule *Availability* pour atteindre les objectifs que nous nous étions fixés pour notre méthodologie. Ces quatre méthodologies sont *IT Infrastructure Library* (*ITIL*) du *CCTA*, *Monitoring Design* (*MD*), *Event Monitoring Design* (*EMD*) et *Tivoli Implementation Methodology* (*TIM*), toutes trois d'IBM. *ITIL* est une méthodologie destinée à organiser les processus de l'entreprise. Elle est assez imposante de par sa taille et est décrite au sein de 34 livres. Les méthodologies IBM sont, quant à elles, toutes propriétaires.

Dans la troisième partie, nous décrirons les postulats de départ qui vont servir de fondement à la nouvelle méthodologie. Dans ces choix, qualifiés de stratégiques parce qu'ils vont conditionner tout le design de la méthodologie, nous allons établir des règles de supervision. Celles-ci vont fixer les limites et contraintes des supervisions et régir les droits et les devoirs de toute personne impliquée dans un projet de supervision. Nous allons également élaborer une architecture, des profils ainsi que des niveaux de supervision qui devront être utilisés dans tout projet de supervision.

Une fois ces choix établis, nous pourrons nous lancer dans l'élaboration d'une méthodologie. La quatrième partie va donc s'atteler à la décrire. Celle-ci se divisera en quatre phases (Interview client, Etude de faisabilité, Développement, Tests et acceptation, Déploiement et suivi de la solution), elles-mêmes divisées en activités. Ces dernières peuvent être des réunions, des séances de travail ou des présentations. Tout au long des phases, et donc des activités, seront produits des documents qui pourront servir de base à une phase ou activité suivante. Cette méthodologie qui devra atteindre les objectifs que nous nous étions fixés dans la première partie, sera construite sur base des idées intéressantes que nous avons retenues dans les différentes méthodologies étudiées et à partir de notre expérience personnelle dans le métier de la supervision.

Enfin, dans la dernière partie de cette étude, nous verrons comment réagit la méthodologie dans les trois grands cas de projet de supervision que rencontre la cellule *Availability*, à savoir la supervision de plates-formes techniques, d'applications existantes et de nouvelles applications.

Pour ce faire, nous verrons les particularités d'utilisation de la méthodologie pour les deux premiers cas tandis que pour les nouvelles applications, nous établirons la manière dont chaque phase et activité de la méthodologie doivent s'intégrer dans le cycle de développement d'une application. Nous prendrons comme exemple le cycle de développement en V qui semble le plus utilisé au sein du département informatique de l'entreprise.

1 Cadre d'étude

1.1 FORTIS

Sur son marché domestique, le Bénélux, FORTIS est l'un des plus grands prestataires de services financiers. Il y offre une large gamme de services financiers via divers canaux de distribution. Dans le reste de l'Europe, aux Etats-Unis et en Asie, FORTIS vise des segments de marché spécifiques. Fin 1998, le total du bilan de FORTIS s'élevait à EUR 338 milliards et le résultat avant impôts à EUR 2,5 milliards. FORTIS regroupe plus de 300 entreprises et emploie plus de 59 000 personnes.

FORTIS se compose de FORTIS Banque et de FORTIS Insurance. Cette dernière, composée notamment d'AG 1824 et d'AMEV, regroupe les activités d'assurance. Alors que FORTIS Banque vend l'arsenal complet des services financiers par le biais de canaux de distribution propres, FORTIS Insurance fait appel à un réseau d'intermédiaires indépendants (courtiers et autres canaux).

1.2 FORTIS Banque

1.2.1 Introduction

FORTIS Banque existe juridiquement depuis le 24 juin 1999. Cette nouvelle banque regroupe les activités de la Générale de Banque et de la CGER en Belgique et de MeesPierson, de la Generale Bank Nederland et de VSB Bank aux Pays-Bas, ainsi que les activités de leurs filiales.

FORTIS Banque occupe une position de leader dans le Bénélux, son marché intérieur. Elle est aussi très active en Europe et dans les principaux pays d'Amérique, d'Asie et d'Afrique. La stratégie de FORTIS Banque s'articule autour de métiers spécialisés, ce qui se traduit dans son organisation. Sur le plan géographique, cette stratégie repose sur trois cercles concentriques.

1.2.2 Organisation

L'organisation de Fortis Banque se présente comme suit :

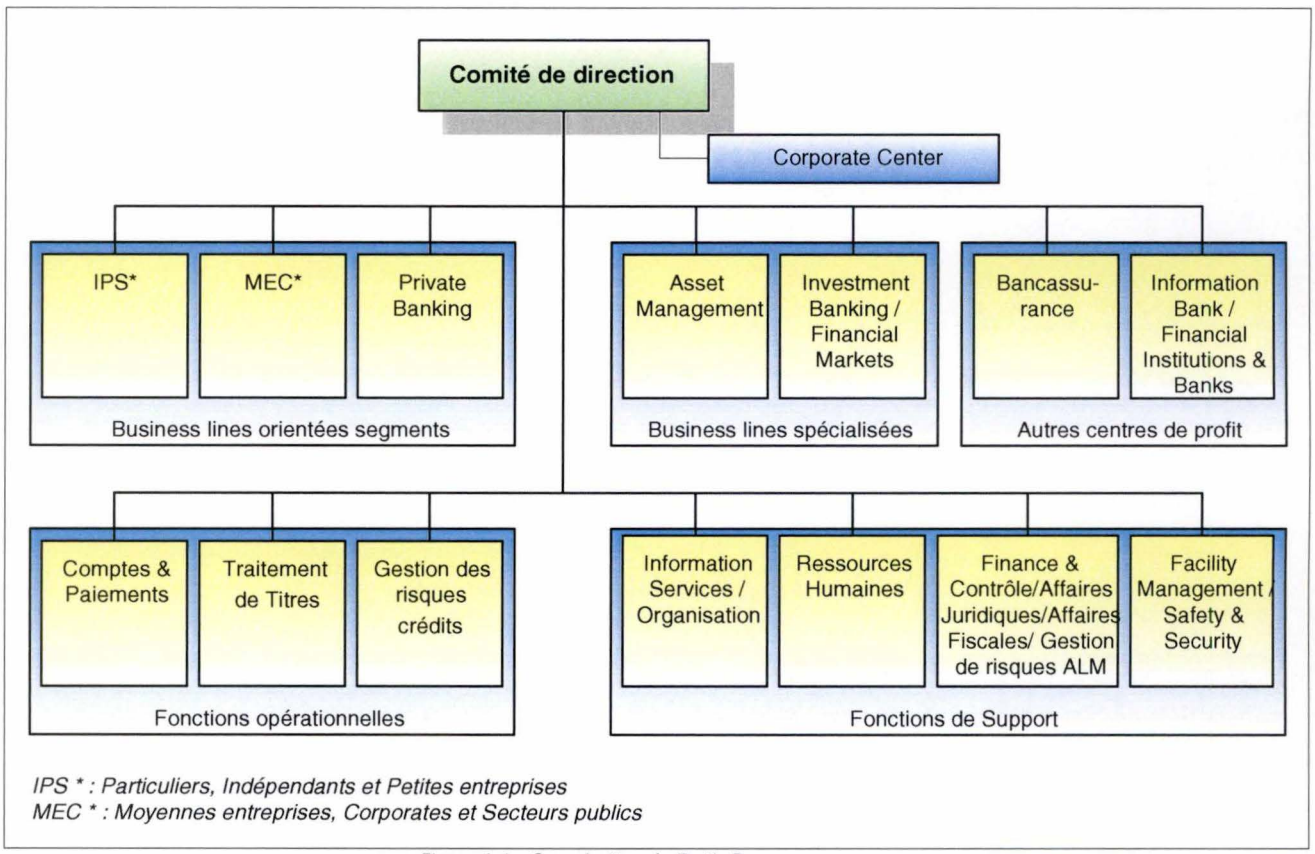


Figure 1-1 Organisation de Fortis Banque

Corporate Centre

Le *Corporate Centre* a pour mission d'assister le Comité de Direction dans ses décisions, dans la planification globale et le contrôle de l'entreprise ainsi que dans la réalisation du processus d'intégration.

Bancassurance

Bancassurance est le pôle de développement de la bancassurance, principalement pour les particuliers, mais également pour les PME et d'autres clients spécifiques liés au réseau bancaire.

Asset Management

Bien que FORTIS se situe dans le top 15 européen des gestionnaires de fonds, elle est encore de taille relativement "réduite".

Gestion des risques - Crédits

Une des activités de base de la banque est l'octroi de crédits aux particuliers, aux entreprises et aux institutions publiques. Ces différents segments de clients sont, au sein de FORTIS Banque, servis par les business lines IPS et MEC.

MEC (Moyennes entreprises et *Corporate*)

MEC est au service des moyennes et grandes entreprises dont les besoins sont pris en charge par des spécialistes. La *business line* MEC a pour mission d'offrir un service bancaire global, partout où les entreprises souhaitent mener et développer leurs activités, en Europe et au-delà.

Ressources Humaines

La fonction de support Ressources Humaines a pour mission globale de gérer, en collaboration avec les responsables des différents métiers, le personnel de FORTIS Banque.

IPS (particuliers, professionnels et petites entreprises)

La ligne de produit IPS s'adresse à trois grands segments de clients à savoir, les particuliers, les professionnels et les petites entreprises.

Les orientations principales d'IPS sont :

- ✓ Mener une approche commerciale et offrir un niveau de service différencié selon les segments de clients, en fonction de leurs besoins et de leur niveau de rentabilité.
- ✓ Avoir une politique de distribution différenciée et assurer un confort maximal pour les clients.
- ✓ Investir dans les nouveaux concepts de distribution tant dans le *non-bricks* (*self-banking*, *phone-banking*, *electronic-banking*, ...) que dans les concepts de distribution physique (agences dans les centres commerciaux, agences étudiants, etc.).
- ✓ Améliorer l'équipement de nos clients et mettre l'accent sur la fidélisation des clients existants, en développant une approche clients axée sur les besoins et le cycle de vie.
- ✓ Promouvoir les activités de vente, de conseil et de prospection.
- ✓ Maîtriser les coûts de distribution.

Facility Management / Security & Safety

Facility Agences et Facility Grands Bâtiments sont responsables de la gestion, de la mise à disposition, de l'entretien et de l'agencement des bâtiments. Facility Management prend également en charge les travaux de construction, de transformation, le renouvellement des enseignes, des logos, etc. planifiés dans le cadre de FORTIS Banque.

Information Services / Organisation

Information Services (IS) définit et développe les moyens informatiques qui permettent d'offrir aux clients des produits financiers concurrentiels.

Private Banking

La ligne de produit *Private Banking* propose une gestion de patrimoine intégrée aux particuliers, fondations et associations ayant plus d'un million d'euros à investir.

Comptes et Paiements

La ligne de produits "Comptes et Paiements" a notamment en charge la gestion des fichiers clients (particuliers et sociétés), la gestion des comptes (à vue, épargne, à terme), la comptabilité clients (intérêts, frais, etc.) et les cartes de paiement.

Traitement des Titres

La fonction opérationnelle Traitement des Titres a pour mission d'assurer, dans les délais, un traitement efficace et correct de toutes les transactions sur titres et options générées par les business lines de FORTIS Banque. Elle définit aussi la politique en matière de titres et représente la banque dans les organismes bancaires et internationaux pour cette matière.

Information Bank / Financial Institutions and Banks

Information Bank traite automatiquement, au niveau mondial, les transactions monétaires ou relatives aux marchés des capitaux, ainsi que les flux d'informations qui leur sont associés, pour le compte d'investisseurs professionnels, qu'ils se trouvent dans ou en dehors du circuit bancaire (gestionnaires de fonds, fonds de pension, etc.).

Finance et Contrôle / Affaires Juridiques / Affaires Fiscales / Gestion des Risques-ALM

La fonction de support Finance et Contrôle a pour mission la production de données pertinentes et fiables, l'initiation et la coordination de l'organisation administrative de la banque. Affaires Juridiques est le conseiller juridique de la banque. Il fournit des services à la banque et à ses employés. Affaires fiscales assume la responsabilité globale en matière fiscale pour les diverses entités de FORTIS Banque.

Investment Banking / Financial Markets

Investment Banking couvre un ensemble d'activités spécialisées comme les activités sur les marchés des capitaux ou dans le domaine des montages financiers. Le *Financial Markets* est actif sur les marchés des changes, monétaires, obligataires et des matières premières.

1.2.3 Quelques dates

- 1990 FORTIS naît du mariage des compagnies d'assurances belge AG et de la néerlandaise AMEV-VSB.
- 1993 L'institution publique CGER est privatisée : FORTIS en acquiert d'abord la moitié des actions et, plus tard, la totalité.
- 1995 La Générale de Banque rachète le Crédit Lyonnais Bank Nederland : la Generale Bank Nederland est née.
- 1997 FORTIS rachète MeesPierson au groupe ABN-AMRO.
- 1998 La Générale de Banque devient partie intégrante de FORTIS.
- 1999 FORTIS Banque naît de la fusion de la Générale de Banque et de la CGER en Belgique, et de la Generale Bank Nederland, de la VSB Bank et de MeesPierson aux Pays-Bas.

1.3 Le département informatique

1.3.1 Introduction

Le département informatique a en charge la gestion de l'infrastructure informatique de la FORTIS Banque. C'est lui qui assure également la compatibilité et l'intégration des outils et des infrastructures des autres entités du Groupe.

Cela implique aussi bien des fonctions consistant à :

- ✓ Développer les applications bancaires.
- ✓ Gérer et faire évoluer les Systèmes (matériel, systèmes logiciels et réseaux).
- ✓ Veiller à l'exploitation quotidienne optimale des services informatiques et des moyens de télécommunication.
- ✓ Pourvoir au support logistique et aux conseils nécessaires au développement et à la gestion de projets (notamment en matière de méthodologie, d'outils et de formation).

1.3.2 Organisation

Le département informatique est organisé comme suit :

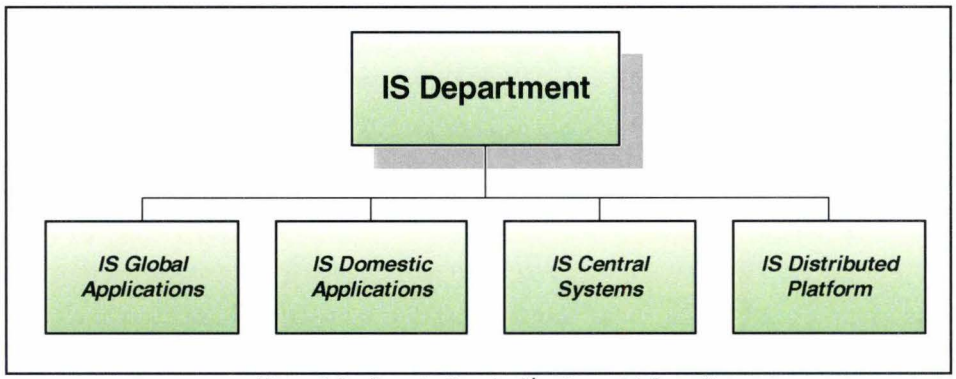


Figure 1-2 Organisation du département informatique

IS Global Applications

IS Global Applications est en charge des applications informatiques pour les branches *Financial Markets*, *Investment Bank*, MEC, *Private Banking* et *Asset Management*.

IS Domestic Applications

IS Domestic Applications est en charge des applications informatiques pour les agences, *Non-Brick Banking*, *Bankinsurance* et Credit.

IS Central Systems

IS Central Systems est responsable de toute l'activité liée aux systèmes centraux tant au niveau des systèmes que des applications. Cette responsabilité couvre aussi bien la sécurité, le planning des tâches informatiques que la supervision des systèmes et des applications informatiques.

IS Distributed platforms

Le champ d'action de la division *Distributed platforms* comprend toute technologie informatique qui ne tombe pas sous le domaine des systèmes centraux. Cette division comprend trois départements : *Distributed Systems*, *Office Automation* et *Staff*.

1.4 Le département ESM : Enterprise Systems Management

1.4.1 Responsabilités

Le département ESM fait partie de la branche *IS Central Systems* du département informatique.

Ses responsabilités sont définies sur trois axes :

- ✓ Recueillir et centraliser les besoins des utilisateurs, tant internes qu'externes, en matière de gestion de systèmes.
- ✓ Discuter, avec les départements techniques, des meilleures solutions à mettre en place pour répondre à ces besoins.
- ✓ Offrir des solutions standards permettant une supervision de bout en bout d'applications et de plates-formes techniques. Ces solutions seront basées autant que possible sur la famille de produits *Tivoli*.

1.4.2 Organisation

Le département ESM se divise en trois cellules :

- ✓ *Architecture & Deployment* : définit l'architecture *Tivoli* au sein de FORTIS Banque.
- ✓ *Availability* : fournit et gère les outils de supervision pour les applications et les plates-formes techniques.
- ✓ *Management tools* : met à disposition des *Help Desk* des outils leur permettant de fournir une réponse adéquate aux problèmes rencontrés par les utilisateurs.

1.4.3 La cellule Availability

La cellule *Availability*, composée d'un chef de cellule et de huit programmeurs, fournit les outils de supervision pour les applications et les plates-formes techniques. Elle développe également les solutions de supervision utilisant ces outils. Ceci en fait non seulement une cellule technique mais aussi de développement plus classique.

La cellule est donc amenée à développer des solutions de supervision pour les autres départements informatiques. Ces départements seront, tout au long de cette étude, mentionnés sous le nom de client.

Enfin, les projets de supervision qu'est appelée à mener la cellule *Availability*, sont de taille et de type très variable. Cela peut aller de la supervision d'un simple programme à la supervision d'un système d'information complet en passant par la supervision d'un middleware ou d'un système d'exploitation. Tout au long de notre propos, nous utiliserons l'expression système d'information ou encore son acronyme SI pour indifféremment faire référence à n'importe quel type ou taille de système dont fait l'objet la supervision.

1.5 Notions de supervision

1.5.1 Introduction

Le monde de la supervision de systèmes ou d'applications informatiques est un monde très particulier en ce sens qu'il s'appuie sur des outils très spécifiques et les développements bâtis autour de ces outils pour assurer la supervision le sont encore davantage. Dans cette section, nous proposons donc de décrire brièvement ce monde afin de mieux en comprendre les particularités et les difficultés.

1.5.2 Les outils de supervision

Pour assurer la supervision de systèmes ou d'applications informatiques, le département ESM[1] utilise le produit *Tivoli Enterprise*, communément appelé *Tivoli*, de la société Tivoli Systems Inc. Celle-ci est, depuis 1996, une filiale à 100 % d'IBM dont le quartier général est situé à Austin, Texas, Etats-Unis. Dans ce paragraphe, nous décrirons très brièvement cet outil.

Le fondement de l'architecture de *Tivoli* est un logiciel distribué, orienté objet[2], appelé *Tivoli Framework* ou plus simplement *framework*. Celui-ci fournit les services de base tels que les communications, la présentation ou la sécurité. Il offre également des facilités de transfert de fichiers et d'exécution à distance de commandes sur des systèmes distribués. En plus du *framework*, *Tivoli* fournit une suite de produits dans les disciplines de gestion telles que le déploiement, la gestion de disponibilité et de la sécurité. La plupart de ces produits utilisent les services fournis par le *framework*. Pour faire partie de *framework* et donc pouvoir utiliser les services qu'il offre, un programme appelé agent doit être installé sur la plate-forme.

Dans la discipline de la gestion de la disponibilité qui concerne cette étude, les deux produits les plus importants de *Tivoli* sont :

- ✓ *Tivoli Distributed Monitoring (TDM)*. *TDM* permet de superviser les ressources et les applications d'un système, d'exécuter d'éventuelles actions correctrices et d'informer les administrateurs de problèmes potentiels. Ce produit comprend un groupe de petits programmes, appelés moniteurs, dont l'unique fonction est de tester l'état de fonctionnement ou la charge (occupation disponible, charge de processeur, taille de fichiers, ...) d'une ressource. *TDM* comprend également un outil, appelé adaptateur journal ou plus simplement adaptateur, destiné à détecter dans les journaux des systèmes ou des applications, l'apparition de messages d'erreur.
- ✓ *Tivoli Enterprise Console ou T/EC*. *T/EC* collecte, traite des alarmes provenant de systèmes d'exploitation, d'applications, de réseaux ou de bases de données. Il permet également, en réponse à ces alarmes, d'exécuter automatiquement d'éventuelles actions correctives. Il représente le point central de contrôle pour les alarmes provenant de toutes ces sources. *Tivoli Enterprise Console* utilise les moniteurs vus précédemment pour collecter l'information, un serveur central pour traiter cette information et des consoles d'alarmes distribuées pour la présenter aux administrateurs des systèmes.
- ✓ *Tivoli Service Desk ou TSD*. *TSD* est une suite d'applications qui permet une gestion de matériel informatique (*Asset Management*) de leurs problèmes (*Problem Management*) et de leurs changements (*Change Management*). Toutes ces applications collaborent de telle manière qu'il est possible de gérer, pour

[1] Voir Le département ESM : Enterprise Systems Management page 18.
[2] Basé sur CORBA 1.1, Common Object Request Broker.

toute l'entreprise, tout le cycle de vie du matériel informatique ainsi que les activités de suivi des problèmes et des demandes de changement liées à ce matériel.

Parmi tous les objets que contient l'architecture de *Tivoli*, on peut en citer deux qui vont nous intéresser pour la suite de cette étude, à savoir :

- ✓ Les *Tasks* (tâches) sont des programmes de petite taille, exécutés à partir d'un point central et dont les paramètres (plate-forme cible et éventuels paramètres de fonctionnement) sont à fournir lors de leur exécution. Ces tâches permettent d'intervenir, à partir d'un point central, directement sur les ressources supervisées afin de résoudre un problème ou de manipuler une ressource. Les tâches les plus courantes sont l'arrêt, le démarrage et l'interrogation (demande d'état de fonctionnement, de configuration, ...) d'une ressource (programme, connexion, ...).
- ✓ Les *Jobs* (tâches personnalisées) sont des tâches pour lesquelles tous les paramètres ont été fixés à l'avance. Les *Jobs* sont conçus soit parce qu'elles sont fréquemment exécutées sur une ressource particulière, soit pour permettre une exécution rapide sur des ressources jugées critiques par les responsables techniques.

1.5.3 Décomposition d'une supervision

La supervision d'un système ou d'une application est assurées par les outils de supervision *Tivoli* décrits dans le paragraphe précédent[3] et dans lequel le lecteur trouvera la définition des termes utilisés dans ce pararaphe. Ce processus, que nous allons décrire dans le détail, peut être représenté par le graphique suivant :

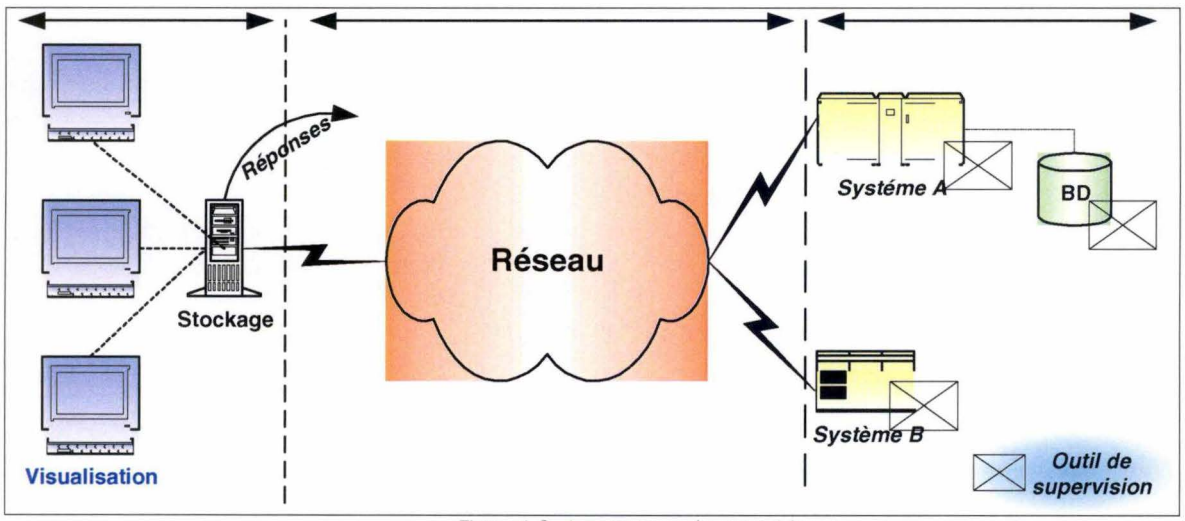


Figure 1-3 Le processus de supervision

La supervision de systèmes ou d'applications informatiques peut être découpée en trois grandes activités distinctes : la surveillance, l'acheminement et le traitement.

La surveillance

La surveillance d'un système ou d'une application consiste à contrôler à intervalles réguliers l'état de différentes ressources ou le contenu de journaux électroniques de ce système ou de cette application. Pour surveiller l'état d'une ressource, la cellule *Availability* dispose des moniteurs. Ces moniteurs couvrent approximativement 70 % des

[3] Voir Les outils de supervision page 19.

besoins en supervision. Pour les 30 % restant, il est nécessaire d'écrire un moniteur approprié. On parle alors dans ce cas de moniteurs personnalisés. En ce qui concerne la surveillance des journaux des systèmes ou des applications, la cellule utilise les adaptateurs. Tant les moniteurs que les adaptateurs demandent, pour leur fonctionnement, la présence d'un agent *Tivoli*. Si pour quelque raison, technique par exemple, la présence d'un agent n'est pas possible, des moniteurs spécifiques doivent alors être écrits soit par le client, soit par la cellule *Availability*. Ces moniteurs envoient alors les alarmes en faisant appel à un petit programme *Tivoli* ne nécessitant pas la présence d'un agent.

Il existe deux types d'alarmes :

- ✓ Des alarmes d'incident qui marquent le début ou l'aggravation d'un problème.
- ✓ Des alarmes qui indiquent la fin d'un problème. Ces alarmes sont communément appelées alarmes positives.

Enfin, une surveillance peut être complétée par des tâches et/ou des tâches personnalisées. Les programmes de supervision (moniteurs et adaptateurs) installés sur une ressource ainsi que les tâches de manipulation de cette ressource constituent le profil de supervision de cette ressource.

L'acheminement

Il existe deux modes d'acheminement des alarmes :

- ✓ Le mode sécurisé utilise le *framework* pour l'acheminement des alarmes. Celui-ci nécessite donc la présence d'un agent à la source mais en revanche l'arrivée des alarmes au serveur d'alarmes est garantie.
- ✓ Le mode non sécurisé consiste à envoyer les alarmes via le réseau local. L'arrivée de ces alarmes au serveur d'alarmes n'est donc pas garanti.

Il est important de signaler que, de par la multitude des chemins possibles dans le réseau, l'ordre d'arrivée des alarmes au serveur d'alarmes n'est jamais garanti et ce, quel que soit le mode d'acheminement choisi.

Le traitement

Le traitement des alarmes regroupe les processus de stockage, de visualisation et de réponse aux alarmes.

Le stockage des alarmes est assuré par la base de données du serveur *T/EC*.

La visualisation des alarmes est possible grâce aux consoles *T/EC*. Ces consoles permettent de visualiser en temps réel les alarmes reçues. Ces alarmes sont regroupées sur ces consoles au sein d'un ou plusieurs groupes appelés vues.

Il existe deux types de vues :

- ✓ Les vues techniques regroupent les alarmes originaires d'un même type de composants. Par exemple, les alarmes remontant des serveurs *NT*, *Unix*, des serveurs *Exchange*, des bases de données *Oracle*, ...
- ✓ Les vues applications regroupent les alarmes issues de tous les composants faisant partie d'un même système d'information (SI) et pour lesquels une supervision existe. Pour constituer ces vues, on construit une vue à laquelle on applique des filtres sur le contenu des champs des alarmes (par exemple, sur le nom d'une plate-forme, d'une base de données ou d'une connexion). On ne laisse ainsi apparaître que les alarmes dont les champs contiennent une valeur identique aux filtres. Par exemple, la vue application du SI "*Web Banking*" regroupe les alarmes venant du serveur *NT* XYZ, du serveur *Unix* ABC, d'un serveur Internet INT1 et d'une base de données *Oracle* DB0001, ... qui composent ce SI.

Une alarme, visible à la console, peut prendre trois états différents :

- ✓ Ouverte lorsque l'alarme concerne un problème actif.
- ✓ Acquittée lorsque la résolution du problème est prise en charge par un niveau de support.
- ✓ Clôturée lorsque le problème lié à l'alarme est résolu. La clôture d'une alarme peut être manuelle ou automatique. Cette dernière est provoquée par la réception d'une alarme positive.

Le processus de réponse aux alarmes se divise en deux parties distinctes : la corrélation et les actions automatiques.

La corrélation est le processus des décisions à prendre en fonction de l'alarme reçue. Cette décision peut être par exemple :

- ✓ Stocker ou non l'alarme. La décision de ne pas stocker une alarme peut être prise si par exemple, l'alarme reçue est une alarme positive ou si une alarme indiquant un problème plus grave est déjà présente.
- ✓ La clôture d'une ou plusieurs alarmes stockées. Il s'agit d'une décision classique lors de la réception d'une alarme positive.
- ✓ L'augmentation de sévérité d'une alarme suite à la réception d'une alarme indiquant un problème plus grave. Ce phénomène est appelé **escalade**.
- ✓ Le déclenchement d'actions automatiques telles que des redémarrages de ressources sur la plate-forme d'où est originaire l'alarme.

Les actions automatiques sont très variées. On peut citer à titre d'exemple les deux plus utilisées, à savoir l'exécution de tâches directement sur le composant d'où est originaire l'alarme et la génération d'un **ticket d'incident**. Ce dernier, qui décrit le problème dont l'alarme fait état, est envoyé et stocké dans l'application de gestion des problèmes ***TSD***[4]. Un ticket peut prendre les états suivants :

- ✓ Ouvert lorsqu'il est généré;
- ✓ Transféré lorsqu'un niveau de support le transmet à un autre;
- ✓ Clôturé lorsque le problème lié au ticket est résolu ou fermé par un utilisateur.

On remarque aisément que la clôture d'un ticket provoque également la clôture de l'alarme associée à ce ticket et *vice versa*.

1.5.4 Caractéristiques des développements de surveillance

Les développements liés à l'activité de surveillance dans le cadre d'une supervision présentent certaines caractéristiques qu'il est intéressant de relever. On peut notamment citer :

- ✓ Le langage de programmation
  De par l'hétérogénéité des plates-formes susceptibles d'être surveillées, un langage compilé est à proscrire. On utilisera donc un langage interprété et universel[5], c'est-à-dire supporté par un maximum de plates-formes différentes.
- ✓ La complexité
  Ces programmes sont exécutés en arrière-plan et ne disposent d'aucune interface utilisateur, aucun accès à des bases de données, ... Ils sont donc d'une complexité très limitée.

---

[4] Voir Les outils de supervision page 19.
[5] Des langages tels que *Java*, *Shell* ou *PERL* sont les plus utilisés.

- ✓ La taille

  Ce sont des programmes très simples qui se limitent à tester la disponibilité ou le bon fonctionnement d'une ressource. Ils sont donc de taille très restreinte, une cinquantaine de lignes en moyenne.

1.5.5 Conclusion

Comme nous l'avons vu, les projets de supervision sont très variés et touchent un grand nombre de domaines de l'informatique. Ainsi, il n'est pas rare de développer une supervision mêlant des systèmes du monde distribué, des systèmes centraux, différents systèmes de gestion de bases de données et des systèmes transactionnels. Les personnes amenées à travailler sur ce genre de projets doivent donc avoir une culture générale du monde informatique assez développée. Il est évident qu'il n'est pas possible d'être spécialiste dans tous les domaines et c'est pourquoi il est important de pouvoir dialoguer avec les personnes possédant l'expertise des différents systèmes.

De plus, l'activité de surveillance est assurée par des outils et des programmes installés directement sur les composants à superviser. Cela soulève deux problèmes :

- ✓ Le risque de mettre en péril le bon fonctionnement ou l'intégrité de l'application ou la plate-forme à superviser. La supervision devra donc, comme toute application, subir le processus de validation mis en place par les responsables de cette application ou plate-forme.
- ✓ L'installation des moniteurs sur ces composants devra être synchronisée avec la planification des cycles d'installation ou de mise à jour de ces mêmes composants.

Point positif, de par leurs caractéristiques, les langages de programmation utilisés pour les moniteurs vont rendre minimes les besoins en formation des programmeurs, limiter le nombre des sources et présenter des cycles de développement et de tests très courts.

1.6 Support aux utilisateurs

1.6.1 Introduction

Dans le paragraphe précédent, nous avons fait référence à plusieurs reprises au support. Cette notion fait référence à l'organisation mise en place pour assurer le support informatique aux utilisateurs. Nous allons, au cours de ce chapitre, détailler les différents niveaux de support mis en place au sein du département informatique de Fortis Banque.

A noter que tout au long de ce travail, il sera indifféremment fait référence aux mots "support" ou "support technique" pour désigner le support aux utilisateurs.

1.6.2 Structure du support

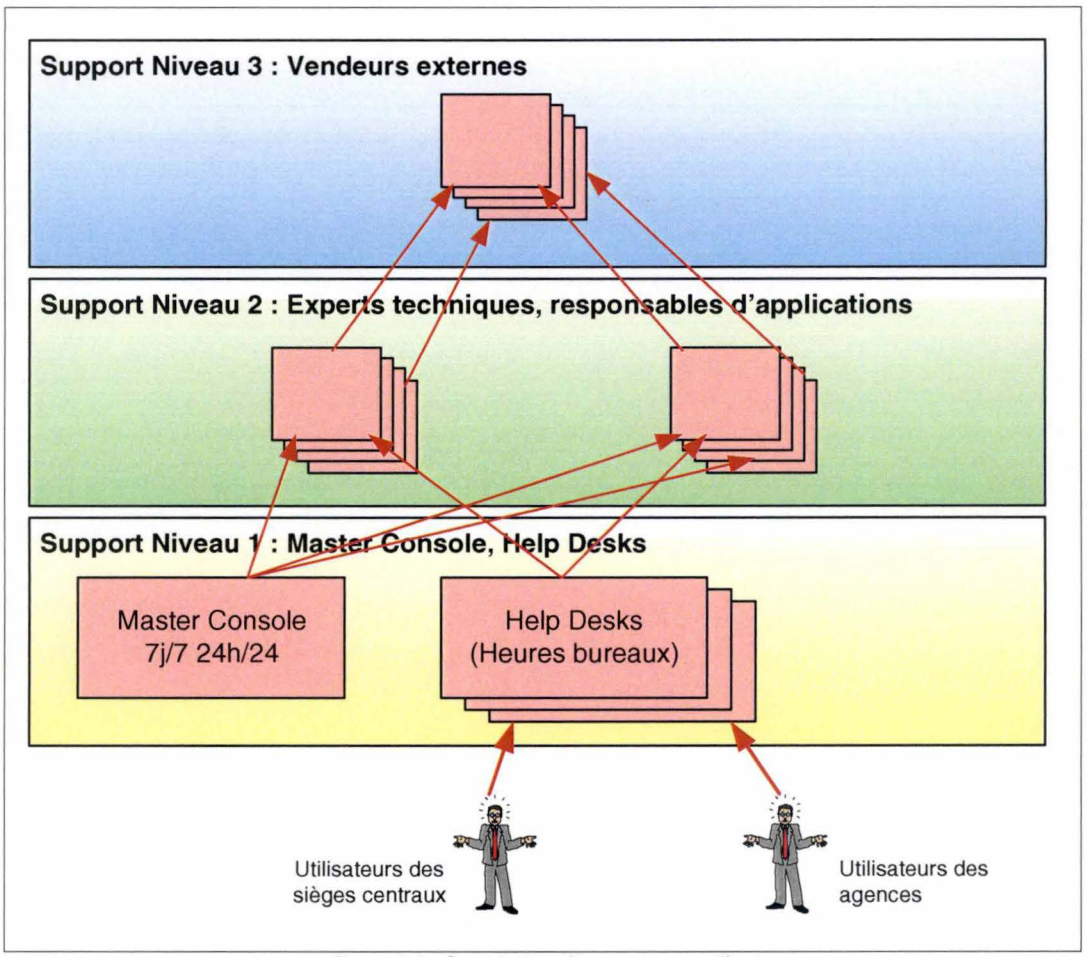


Figure 1-4 Organisation du support aux utilisateurs

Le support aux utilisateurs des sièges centraux et des agences est centralisé au sein de services spécialisés. Il s'organise sur 3 niveaux :

- ✓ Le niveau 1 regroupe les différents *Help Desk* et la *Master Console*.
- ✓ Le niveau 2 est formé de tous les départements responsables des applications et des plates-formes techniques.
- ✓ Le niveau 3 regroupe les sociétés de logiciels et les constructeurs informatiques.

1.6.3 Support de niveau 1

Le niveau 1 est le niveau de support de première ligne. Il se compose de deux entités distinctes :

- ✓ Les *Help Desk* sont avertis d'un problème par la réception d'un ticket d'incident ou par un appel d'un utilisateur. Ils tentent dans la mesure du possible de résoudre le problème. Ils fournissent donc une assistance réactive aux problèmes. Dans la majeure partie des cas, les *Help Desk* assurent le support uniquement pendant les heures de bureau. Pour assurer cette mission, les *Help Desk* utilisent principalement l'outil *TSD*[6].
- ✓ La *Master Console* opère une surveillance pro-active de l'infrastructure informatique, 24 heures sur 24 et 7 jours sur 7. La *Master Console* dispose de plusieurs consoles techniques pour superviser l'infrastructure informatique mais la principale est une console *T/EC* fournie par la cellule *Availability* appelée ***T/EC* Entreprise**. Sur cette console n'apparaissent que les alarmes marquant une indisponibilité ou significative d'une dégradation de service de l'infrastructure informatique. A chaque alarme est associée l'adresse d'une page dans l'Intranet du département informatique où la *Master Console* peut trouver l'information nécessaire à la résolution du problème ainsi qu'une liste de personnes à contacter si nécessaire.

Si le support de niveau 1 ne parvient pas à résoudre un problème, il transfert le problème au niveau 2. Ce transfert s'effectue par l'envoi du ticket relatif à l'incident à l'équipe de support de niveau 2 concernée.

1.6.4 Support de niveau 2

Le support de niveau 2 est constitué :

- ✓ Des départements techniques, c'est-à-dire des départements responsables des plates-formes techniques (NT, Unix, Mainframe, ...), des réseaux (SNA, X25, LAN, ...), des bases de données (*Oracle*, *Informix*, *DB2*, ...) et des middleware (*MQSeries*, *Exchange*, *Web Server*, ...).
- ✓ Des responsables des applications.

Pour ce support, deux modes de travail sont généralement utilisés :

- ✓ Réactif : réaction aux problèmes suite à un appel du niveau 1 ou à la réception d'un ticket d'incident.
- ✓ Pro-actif : surveillance directe de l'infrastructure via une console contenant des vues techniques pour les services techniques ou des vues applications pour les responsables des applications.

Ces deux modes de travail peuvent se combiner selon les exigences de disponibilité ou les besoins des responsables. Par exemple, il n'est pas rare de voir le niveau 2 assurer un support pro-actif pendant les heures de bureau et un support réactif en dehors de ces heures.

1.6.5 Support de niveau 3

Le support de niveau 3 est composé de toutes les firmes de matériel ou de développements externes à la société (Telinfo, Microsoft, Cable Print...). Ils sont appelés par le niveau 2 pour des interventions techniques ou des demandes de corrections d'applications.

---

[6] Voir Les outils de supervision page 19.

1.7 Audit

1.7.1 Introduction

La cellule *Availability* est responsable de la mise en place d'outils et du développement d'applications de supervision de plates-formes techniques et d'applications critiques pour la banque. Depuis sa création, la cellule rencontre d'énormes difficultés pour répondre aux exigences de ses clients tant au niveau de la qualité des solutions qu'elle propose qu'au niveau du respect des délais. A titre d'exemple, pour l'année 2000, **30** % des projets prévus pour l'année ont été réalisés et parmi ceux-ci, seulement **10** % l'ont été dans les délais impartis mais avec une qualité fort inégale. Pour connaître les raisons de ce manque de productivité, un audit sur le mode de fonctionnement de la cellule et ses processus de développement s'impose.

On peut regrouper les problèmes que rencontre la cellule *Availability* en cinq groupes :

- ✓ Humain
- ✓ Organisationnel
- ✓ Standardisation
- ✓ Gestion de projet
- ✓ Délais

1.7.2 Problèmes humains

La cellule *Availability* n'échappe pas aux mouvements de personnel que connaît actuellement le secteur informatique belge. A titre d'exemple, pour l'année 2000, pas moins de quatre programmeurs de la cellule ont quitté l'entreprise et ont dû être remplacés. C'est d'autant plus préjudiciable pour la cellule que le temps de formation aux outils de supervision utilisés est long (entre trois et six mois suivant l'expérience de la personne) et qu'une période supplémentaire de deux mois est nécessaire pour la formation aux outils et aux environnements de développement propres à la cellule.

1.7.3 Problèmes organisationnels

La cellule *Availability* connaît également de gros problèmes organisationnels. Pour un projet de supervision, mis à part l'installation et la configuration de nouveaux produits, la moitié de la charge de travail réside dans la collecte des besoins du client et la rédaction des spécifications fonctionnelles et techniques. Or, ces tâches sont principalement dévolues au client et au chef de cellule pour la partie collecte des besoins et au chef de cellule, seul qualifié de la cellule pour ce travail, pour les spécifications. De par le nombre croissant des projets, ce dernier n'a plus le temps de rédiger un cahier des charges et des spécifications valables et la plupart des demandes de développement se transmettent donc oralement aux programmeurs. Ceci amène des remises en question continuelles et des changements parfois importants de spécifications qui entraînent des retards importants dans la planification des projets.

1.7.4 Problèmes de standardisation

Tous les développements de supervision se font au coup par coup suivant la demande. Il n'y a pas de standardisation des profils de supervision. Il arrive donc couramment que deux plates-formes techniques ou deux applications identiques aient des profils de supervision différents.

Enfin, les supervisions de tous les composants d'une plate-forme technique (hardware, système d'exploitation, applications...) sont regroupées au sein d'un même profil de

supervision[7]. Ainsi, le profil de supervision d'une application est différent suivant le système d'exploitation auquel il est destiné. Cela multiplie le nombre de profils pour une même application et rend complexe, voire impossible, la gestion des profils en cas de changement. Finalement, on ne sait plus quels composants sont supervisés et comment ils le sont.

1.7.5 Problèmes de gestion des projets

Les problèmes liés à la gestion des projets sont les plus importants rencontrés par la cellule *Availability*. Ils sont nombreux et se présentent tout au long du cycle de vie d'un projet.

Du côté du client :

- ✓ Incapacité de fournir une topologie complète de son système d'information. Ceci se vérifie surtout pour une vue application.
- ✓ Incapacité d'identifier les composants critiques de son application.
- ✓ Par manque d'expertise, incapacité du client à spécifier ses besoins en matière de supervision.

Du côté de la cellule *Availability* :

- ✓ Incapacité d'expliquer clairement au client ce que l'on attend de lui.
- ✓ Pauvreté, voire inexistence des documents de suivi de projet.
- ✓ Incapacité d'évaluer *a priori* la charge de travail et d'établir un planning.
- ✓ Mauvaise réutilisation des développements existants.
- ✓ Documentation technique inexistante.

Toutes ces lacunes entraînent encore une fois des remises en question continuelles des spécifications, des erreurs dans les définitions des profils et dans les développements, des décalages importants entre les spécifications de départ et le produit final et des lacunes dans la documentation. Finalement, les supervisions sont livrées hors délais et avec une qualité inégale. De plus, la multiplication des profils de supervision accroît de manière importante la charge de travail de la cellule pour la maintenance au détriment de nouveaux développements.

1.7.6 Problèmes de délais

Tous ces problèmes empêchent la cellule de fournir des solutions de supervision dans des délais raisonnables. En fait, il faut compter un délai de plus ou moins six mois pour une vue application, et de quatre pour une vue technique. Vu le nombre croissant de projets en attente, les réalisations annuelles pour la cellule peuvent être considérées comme dérisoires.

[7] Voir Notions de supervision page 19.

1.7.7 Conclusion

Afin de pallier tous ces problèmes, une méthodologie orientée "Supervision" s'avère nécessaire. Il est évident qu'une telle méthodologie ne peut résoudre le problème de manque d'effectifs mais elle peut, par contre, aider à optimiser l'utilisation des ressources disponibles et améliorer la gestion des projets. Cette méthodologie devra permettre une industrialisation des développements destinés aux supervisions et donc une augmentation de la productivité de la cellule *Availability*. Cette méthodologie devra :

① Définir des standards, des règles et une architecture pour la supervision.

② Rendre les profils de supervision des différents composants d'une plate-forme technique indépendants les uns des autres.

③ Définir des standards pour la spécification et la documentation du projet (modèles de documents) afin d'améliorer le dialogue avec le client et le guider d'un bout à l'autre du projet.

④ Minimiser les développements liés aux nouveaux projets de supervision et permettre un réemploi maximum des supervisions existantes.

⑤ Evaluer de manière correcte une planification de développement et aider à respecter les délais.

2 Etude de méthodologies existantes

2.1 Introduction

Avant de se lancer dans la conception de toute pièce d'une méthodologie orientée "Supervision", il nous paraît nécessaire de voir s'il n'existe pas sur le marché informatique une ou plusieurs méthodologies qui pourraient atteindre les objectifs que nous nous sommes fixés.

Il n'existe, dans le domaine de la gestion de la disponibilité de systèmes d'information, qu'un petit nombre de méthodologies. Parmi celles-ci, nous avons choisi d'en étudier quatre qui nous semblent le plus à même d'apporter des réponses aux problèmes que rencontre la cellule *Availability*. Ces méthodologies sont les suivantes :

- ✓ ITIL - Availability Management.
- ✓ Monitoring Design (MD).
- ✓ Event Management Design (EMD).
- ✓ Tivoli Implementation Methodology (TIM).

Il est à noter, que par souci d'exactitude, nous avons gardé la terminologie utilisée par ces méthodologies, c'est pourquoi apparaîtront dans cette étude, nombre de termes ou acronymes anglais.

2.2 ITIL : Availability Management

2.2.1 ITIL

La méthodologie *ITIL, IT Infrastructure Library*, a été développée à la fin des années 80 par le *CCTA*[8], organisme gouvernemental anglais. La totalité de la méthodologie est décrite dans 34 livres regroupés en 5 groupes correspondant aux 5 grands domaines de *ITIL*.

Ces 5 domaines sont :

- ✓ **Service Support** : étudie les questions de livraison et de support des services informatiques appropriés aux exigences de l'organisation. Ces méthodes régissent également l'identification et l'enregistrement de la configuration de l'infrastructure, la communication journalière avec les utilisateurs et la coordination des incidents, des problèmes et des changements.
- ✓ **Service Delivery** : concerne la livraison de services informatiques de qualité et à moindre coût. Cela se fait par la négociation d'accords de niveaux de qualité de service[9] entre le département informatique et les fournisseurs. La méthodologie fournit également les méthodes afin de définir et gérer les fonctions qui doivent être mises en place telles que la gestion des disponibilités (*Availability Management*), de capacité et de planification des charges.
- ✓ **Manager** : fournit aux gestionnaires des méthodes destinées, notamment, à gérer les relations avec les clients et les fournisseurs, à assurer une gestion efficace du personnel informatique et à maintenir une qualité des services.
- ✓ **Computer Operations** : fournit les méthodes nécessaires pour l'installation et l'acceptation de nouveaux sites informatiques. Cela aborde les questions de respect des coûts, des délais, d'exigences, de gestion des risques, ... Elle fournit également des méthodes pour l'exploitation au jour le jour des sites informatiques.
- ✓ **Software Support** : fournit les méthodes pour l'acquisition, la maintenance et les tests des logiciels.

Pour la problématique qui nous occupe, nous allons nous pencher sur la partie *Service Delivery* et plus particulièrement sur le contenu du document consacré à la gestion de disponibilité (*Availability Management*).

2.2.2 Availability Management

L'objectif majeur du document ITIL "Availability Management" est de guider les entreprises dans leur démarche pour atteindre et maintenir le niveau de disponibilité des services informatiques nécessaires pour supporter les activités métiers des clients à un coût justifié.

Ce module se focalise sur les procédures et les systèmes, y compris les spécifications et le contrôle des obligations contractuelles des fournisseurs liées à la disponibilité, nécessaires afin de supporter les besoins en termes de disponibilité décrits dans les contrats de services (SLA).

La gestion de la disponibilité est l'un des éléments clés pour améliorer la qualité des services informatiques fournie aux utilisateurs.

---

[8] Central Computer and Telecommunications Agency, qui en est toujours propriétaire.
[9] Plus couramment appelé SLA, *Service Level Agreements*.

Le CCTA décompose la gestion de la disponibilité selon les axes suivants :

- ✓ ***Disponibilité*** Service disponible pour l'utilisateur lorsque cela est nécessaire.
- ✓ ***Fiabilité*** Capacité d'un service IT à effectuer une fonction sans problème sous certaines conditions spécifiées et sur une période de temps donnée.
- ✓ ***Qualité de service*** Terme contractuel qui détermine les exigences en termes de disponibilité.
- ✓ ***Récupération*** Capacité à restaurer un état opérationnel après une interruption de service.
- ✓ ***Sécurité*** Implications au niveau de la sécurité.

Ce guide définit les étapes de mise en œuvre de la gestion de la disponibilité. Ces étapes, au nombre de trois, sont :

- ✓ Le planning.
- ✓ L'implémentation.
- ✓ La post-implémentation et l'audit.

Le planning intègre les activités suivantes :

- ✓ Etude de faisabilité : établir l'importance de la gestion de la disponibilité par rapport à la problématique métier de l'entreprise.
- ✓ Définition des objectifs et du périmètre.
- ✓ Sensibilisation à la problématique de la disponibilité.
- ✓ Analyse des systèmes en place.
- ✓ Analyse des besoins en termes de disponibilité.
- ✓ Collecte de données détaillées relatives à la disponibilité.
- ✓ Analyse des sources de données de disponibilité.
- ✓ Analyse du stockage des données de disponibilité.
- ✓ Planification de la surveillance de la disponibilité.
- ✓ Planification de la génération de tableaux de bord.
- ✓ Vérification de la capacité à répondre aux besoins.
- ✓ Revues et audits.

L'implémentation est structurée comme suit :

- ✓ Procédures
  - Surveillance de la disponibilité.
  - Génération des tableaux de bord.
  - Prévision de la disponibilité.
    - Evaluation de l'impact des changements.
    - Changements pour améliorer la qualité de service globale.
    - Etudes complémentaires par rapport à la disponibilité.
- ✓ Outils de support aux procédures.
  - Base de données de gestion de la disponibilité.
  - Outils de diagnostic.
  - Outils de modélisation et modèles.

La post-implémentation et l'audit couvrent les aspects de :

- ✓ Revue de post-implémentation.
- ✓ Exploitation et revue continuelle.
- ✓ Production du plan de disponibilité (Availability Plan).
- ✓ Revue de l'efficacité et de la couverture de la gestion de la disponibilité.
- ✓ Audit de la fonction gestion de la disponibilité.

2.2.3 Apports potentiels

ITIL Availability Management couvre tous les domaines présents dans un département informatique. Malheureusement, elle est basée sur une "culture mainframe" construite davantage sur des procédures et des tableaux de bord que sur la mise en place d'une gestion pro-active de la disponibilité. De plus, l'imbrication des modules qui la compose oblige celle-ci à être appliquée pour le département informatique dans sa totalité. Cependant, deux éléments importants nous paraissent intéressants dans le cas qui nous occupe :

- ✓ La phase de planning : permet de définir une ligne de conduite générale pour tous les projets de supervision.
- ✓ La boucle de feed-back (post-implémentation et audit) dont le but est d'évaluer de manière continue l'efficacité et la couverture de la solution de supervision et de l'organisation de la gestion de la disponibilité.

2.3 Event Management Design (EMD)

2.3.1 Domaine d'application

La méthode *Event Management Design*, également connue sous le nom de *Event Management and Correlation Design* : EMCD, est une méthodologie propriétaire d'IBM destinée à :

- ✓ Recenser l'ensemble des alarmes générées par une source et à améliorer leur traitement. Cette source peut être un composant technique (routeur réseau ou système d'exploitation), une application ou un middleware.
- ✓ Fournir la matière nécessaire pour une mise en œuvre efficace de la gestion de ces alarmes.

2.3.2 Composants de la méthodologie

Cette méthodologie se déroule en cinq activités ou ateliers, pour reprendre le terme utilisé dans la méthodologie. Ces ateliers sont :

Atelier 1 Définition de règles générales de gestion.

Ces règles ont comme objectif de fixer les limites et les lignes de conduites de la gestion des alarmes. Elles sont définies sous la forme de tableau dont voici un exemple tiré de la littérature [THOENEN D., 2001, page 17] sur la méthodologie :

Policy	To enter the enterprise tier an event must signal a potential or actual reduction in corporate service level that will result in failure to meet committed service level agreements (SLAs).
Rationale	Event processing at the enterprise tier must focus upon those events that will have an impact upon the conduct of business operations.
Implications	Events signaling disruption of services not covered by corporate SLAs will not be processed at the enterprise tier. These events remain the management responsability of lowers tiers.

Figure 2-1 Méthodologie EMD : exemple de règle

Où Policy est l'énoncé, dans des termes simples, de la règle.

Rationale justifie la règle. Il s'agit en fait de la liste des raisons qui ont incité sa formulation.

Implications décrit l'impact qu'aura la règle sur l'environnement.

Toutes ces règles sont regroupées en cinq grandes catégories classées par domaine abordé, du plus général au plus ciblé :

- ✓ Entreprise.
- ✓ Organisation.
- ✓ Outils et plates-formes.
- ✓ Relations extérieures.
- ✓ Plates-formes cibles.

Atelier 2 Sélection des sources d'alarmes.

L'atelier 2 consiste à dresser une liste des composants de l'environnement informatique de manière à déterminer les types de ressources gérées et les sources d'alarmes qui seront pris en compte lors de la conception. L'objectif de cette base est de définir le périmètre détaillé de la gestion des alarmes. Pour ce faire, un tableau de trois colonnes est utilisé :

Component	Event Source	Decision

Où : Component — Est le type de ressource. Par exemple, un routeur de type XYZ.

Event Source — Indique la source des alarmes pour le type de ressource en question. Par exemple, pour un routeur, la source pourrait être des messages *SNMP*[10].

Decision — Est la décision prise pour les alarmes de cette source. Garde-t-on ou ne garde-t-on pas les alarmes générées par la source donnée pour les ateliers suivants ?

Atelier 3 Inventaire des référentiels d'alarmes.

Pour chaque source d'alarmes, chaque type d'alarme qui peut être généré est identifié et documenté sous la forme :

Event ID	Event Name	Event Description	Event Sub-source

Où : Event ID — Identifiant associé à chaque alarme.

Event Name — Dérivé de la documentation du constructeur du matériel ou l'éditeur du logiciel, par exemple le texte du message.

Event Description — Egalement dérivé de même documentation, par exemple l'explication du message tiré de la documentation technique.

Event Sub-source — Optionnel, utilisé lorsque la source peut apporter des éléments complémentaires pour la qualifier, par exemple une version particulière de la source.

Atelier 4 Décision du filtrage des alarmes.

Cette étape détermine :

- ✓ Les alarmes devant être filtrées ou transmises vers le serveur d'alarmes.
- ✓ L'affectation des degrés de sévérité.
- ✓ La répercussion de ces niveaux de sécurité au niveau de l'outil de gestion des alarmes.
- ✓ La localisation du filtrage des alarmes.

Pour ce faire, le tableau suivant est utilisé :

① Event ID	② Event Name	③ Filter at source	④ Filter at Dist Mgr	⑤ Throttle parm	⑥ Event Category	⑦ Event Status	⑧ Enterprise Significant

Où : ① Event ID — Est repris du tableau de l'atelier 3.

[10] SNMP : Simple Network Management Protocol. Protocole d'administration de composants informatiques. Est devenu un standard dans l'administration de réseaux [TANENBAUM A., 1997].

② Event Name — Est repris du tableau de l'atelier 3.

③ Filter at source — Si un message ne présente pas d'intérêt pour la supervision, il doit être filtré le plus tôt possible. L'endroit idéal est à la source même car cela évite d'acheminer et de traiter des alarmes inutiles.

④ Filter at the DM — Si le filtrage à la source n'est pas possible et qu'il existe un outil de gestion des alarmes distribué[11] pour la source d'alarmes concernée, il est possible de spécifier le filtrage à ce niveau.

⑤ Throttle param. — Le *Throttling* est une technique qui permet de gérer les avalanches d'alarmes répétitives par la suppression des alarmes identiques, comptage des alarmes reçues dans un compteur, déclenchement d'une alarme à partir d'un nombre minimum d'erreurs détectées, etc.

⑥ Event category — N'est complété que lorsque des produits spécifiques tels que *HP OpenView* sont utilisés.

⑦ Event status — Idem ⑥.

Ces deux derniers points sont, nous en convenons, hors sujet mais sont cités par souci d'exactitude.

⑧ Enterprise sign. — Indique si l'alarme apporte ou non une information quant à la disponibilité d'un système. Si tel n'est pas le cas, cette alarme est dite non significative et il y a lieu, dès lors, de filtrer cette alarme.

Atelier 5 Analyse pour la corrélation des alarmes.

Dans un premier temps, il s'agit de décider quelles alarmes sont susceptibles d'être en relation ou corrélées avec d'autres. Pour dresser la liste de ces alarmes, la méthodologie s'appuie sur le tableau suivant :

① Event ID	② Event Name	③ Correlation Candidate	④ Autonomous Event	⑤ Duplicate Detect	⑥ ERN Name	⑦ Forward to Trouble Ticketing	⑧ Event Severity Level

Où :

① Event ID — est repris du tableau de l'atelier 4.

② Event Name — est repris du tableau de l'atelier 4.

③ Correlation Candidate — indique si l'alarme est à mettre en relation avec une ou plusieurs autres alarmes parce qu'elle clôture ou indique l'aggravation d'un problème. Cette relation entre des alarmes s'appelle une corrélation.

④ Autonomous Event — indique une alarme qui sera traitée de manière autonome et qui ne fera donc partie d'aucune corrélation.

⑤ Duplicate Detect — détection d'alarmes dupliquées. Dans ce cas, on compte les doublons des alarmes reçues.

[11] Ou Mid-Level Manager : MLM.

⑥ ERN Name — les corrélations sont décrites sous forme de réseaux de relations appelés *Event Relationship Network* (*ERN*). Si l'alarme doit faire l'objet d'une corrélation, on spécifiera ici le nom de l'ERN dans lequel la corrélation sera décrite.

⑦ Forward to T.T. — est-il nécessaire de générer un ticket d'incident à la réception de l'alarme ?

⑧ Event Severity Level — indique la sévérité qui sera associée à l'alarme.

Dans un second temps, on dessine les réseaux des relations qui décrivent les corrélations des alarmes qui ont été déclarées comme candidates à une telle corrélation. Pour ce faire, on distingue quatre types d'alarmes :

- ✓ *Primary (P)* : une alarme joue un rôle primaire si elle fournit de par sa nature une indication directe ou non ambiguë sur l'action à prendre pour résoudre le problème.
- ✓ *Secondary (S)* : une alarme joue un rôle secondaire si elle est toujours externe en terme de choix d'action à prendre dans des situations exceptionnelles. Elle indique, par exemple, l'aggravation d'un problème.
- ✓ *Secondary/Primary (S/P)* : est utilisé pour une alarme pouvant jouer les deux rôles précédents.
- ✓ *Clearing (C)* : est utilisé pour les alarmes marquant la fin d'un incident.

Chaque alarme est représentée par un cercle de couleur différente indiquant son type (P, S, S/P ou C). Des flèches indiquent l'ordre de génération des alarmes pour les alarmes Primary, Secondary et Secondary/Primary et, pour les alarmes Clearing, les alarmes qu'elles clôturent.

La figure suivante représente un exemple d'ERN généré par un routeur réseau tiré également de la littérature[12] de la méthodologie :

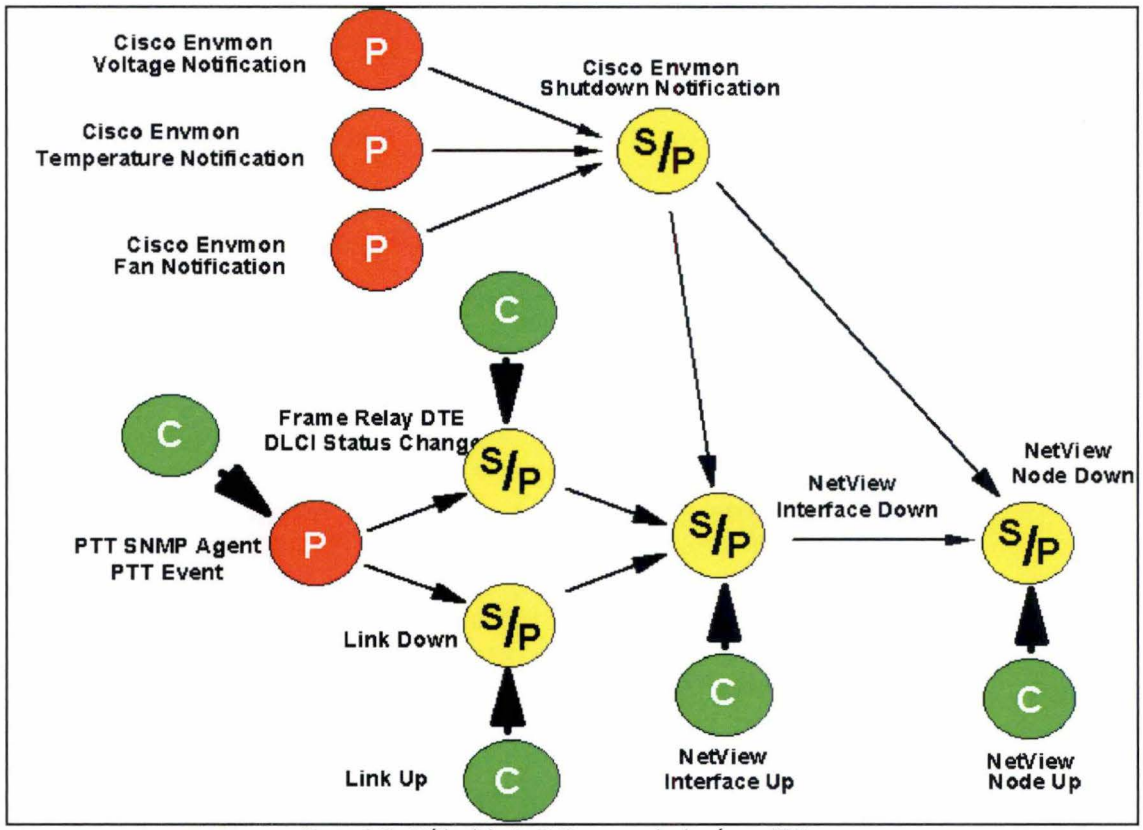


Figure 2-2 Méthodologie EMD : exemple de réseau ERN

Dans cet exemple, on distingue quatre alarmes de type *Primary* en rouge, cinq alarmes *Secondary/Primary* en jaune et cinq *Clearing* en vert. On voit que l'alarme ***Link Up*** de type *Clearing* clôture le problème ***Link Down***. Ce dernier peut être la cause de la génération d'une alarme ***Netview Interface Down*** mais peut également être la conséquence de l'alarme ***PTT SNMP Agent PPT Event***.

2.3.3 Apports potentiels

EMD apporte, d'une part, un moyen efficace de régulariser les projets de supervision grâce aux règles et propose, d'autre part, un formalisme intéressant de documentation des corrélations et permet de s'assurer qu'un traitement approprié sera défini pour chaque alarme. A noter qu'IBM fournit avec cette méthodologie un ensemble d'outils[13] qui permet de générer automatiquement les règles de corrélation à implémenter.

[12] THOENEN D., 2001, page 14.
[13] Malheureusement, ces outils sont propriétaires et donc l'accès en est interdit au grand public.

2.4 Monitoring Design

2.4.1 Domaine d'application

La méthode «Monitoring Design» d'IBM fournit une procédure qui permet d'analyser l'environnement informatique dans le but de mettre en place une surveillance des performances (en temps réel) et de la disponibilité. Cette méthode est principalement utilisée dans le cadre de la conception détaillée de la surveillance à mettre en place par rapport à un service critique. Elle est souvent utilisée en complément à la méthode Event Management Design (EMD).

2.4.2 Composants de la méthodologie

La méthodologie se décompose en quatre grandes activités :

1. Activité d'identification du service.
   Cette activité consolide les informations relatives au service suivantes :
   - ✓ Nom du service.
   - ✓ Propriétaire ou fournisseur du service : département ou individu responsable des performances et de la disponibilité du service.
   - ✓ Description du service : objectifs majeurs et composants de l'architecture.
   - ✓ Besoins en termes de niveaux de service : Décrire les niveaux de service existants. S'il n'en existe pas, décrire ce que serait le niveau minimum acceptable de performance et de disponibilité.
   - ✓ Problèmes identifiés : décrire les problèmes de performance qui ont déjà été identifiés.
   - ✓ Localisation de la documentation complémentaire liée au service.
2. Activité de décomposition du service en éléments gérés.
   Le service est analysé dans ses détails, pour identifier les éléments et les sous-éléments qui constituent les composants du service. On doit s'assurer que cette décomposition pourra s'appuyer sur des moniteurs de base délivrés avec les produits de surveillance. L'approche la plus efficace est de réunir un groupe de travail avec les experts qui peuvent apporter le niveau de détail nécessaire pour tous les composants du service.
3. Documentation du référentiel de surveillance.
   Cette activité a pour but de consolider l'ensemble des moniteurs existants dans les outils et développements spécifiques accessibles pour l'entreprise.
4. Identification des moniteurs et des seuils.
   Cette activité documente les métriques à utiliser, les valeurs de seuil à établir pour chaque niveau de sécurité et les actions à déclencher pour les différents niveaux de sécurité.

2.4.3 Apports potentiels

La procédure pour analyser les caractéristiques de surveillance ainsi que les tableaux associés pourraient faciliter la collecte des spécifications de supervision pour les plates-formes techniques. Malheureusement, ces ressources sont propriétaires et donc inaccessibles.

2.5 Tivoli Implementation Methodology (TIM)

2.5.1 Domaine d'application

TIM est une méthodologie utilisée par *Tivoli*[14] pour déployer des solutions d'administration basées sur les propres produits. Dans cette méthodologie, il y a notamment une phase de consolidation des besoins avec, en particulier, des questionnaires se rapportant à la supervision, de manière générale, et par composants applicatifs.

2.5.2 Composants de la méthodologie

Les composants de la méthodologie sont les suivants :

- ✓ Collecte des besoins : utilise des questionnaires standardisés pour la collecte des besoins des utilisateurs. Plusieurs documents de spécifications seront produits à la fin de cette étape.
- ✓ Analyse du système : est le processus destiné à construire l'architecture *Tivoli* qui sera mise en place. Sur base de plusieurs questionnaires, une proposition technique, un design physique et logique ainsi qu'un calcul de coût seront remis.
- ✓ Planification détaillée du projet : planification complète du projet d'implémentation de la solution *Tivoli*.
- ✓ Déploiement : déploiement de la solution.
- ✓ Test : test de la solution.
- ✓ Documentation : mise en place d'un système de documentation.

2.5.3 Apports potentiels

La méthodologie TIM inclut des questionnaires spécifiques pour chacun de ses composants décrits au paragraphe précédent. Ces questionnaires sont très orientés sur les produits *Tivoli* et sont plutôt destinés au dialogue avec les architectes techniques dans le but de définir les vues techniques.

[14] Voir Les outils de supervision page 19.

2.6 Conclusion

Il est certain qu'aucune de ces méthodologies ne répond à elle seule pleinement aux besoins de la cellule *Availability*. De plus, celles-ci sont, dans la majorité des cas, propriétaires et uniquement accessibles aux entreprises par le biais de contrats de consultance. Les informations libres accessibles au grand public sont très limitées, ou dans le cas de ITIL, accessibles via des cours payants ou une série impressionnante de livres (au nombre de 34) dont leur lecture est très ardue sans un minimum de formation. A l'exception de la méthodologie EMD pour laquelle le département ESM a effectué une consultance, il ne nous a pas été possible d'étudier le contenu des documents, des questionnaires et des procédures utilisés par ces méthodologies; par conséquent, seuls quelques concepts peuvent être retenus :

- ✓ L'idée de boucle de feed-back pour l'évaluation de la supervision (ITIL).
- ✓ L'inventaire des sujets abordés dans l'Availability Management (ITIL).
- ✓ L'idée d'utiliser des tableaux pour la spécification de la supervision (MD).
- ✓ La définition de règles (EMD).
- ✓ Le processus de sélection des alarmes (EMD).
- ✓ Le processus de corrélation (EMD).
- ✓ L'idée d'utiliser des questionnaires standardisés pour la collecte des besoins (TIM).
- ✓ L'idée d'un système structuré de documentation pour les projets et la documentation technique (TIM).

3 Choix stratégiques

3.1 Introduction

Avant de se lancer dans une méthodologie, il est nécessaire de définir des règles ainsi qu'une architecture de supervision qui serviront de charpente sur laquelle nous allons nous appuyer pour construire la méthodologie.

Comme nous l'avons vu précédemment[15], la cellule *Availability* rencontre de gros problèmes de standardisation. Certains problèmes peuvent être résolus en fixant une série de postulats de départ qui seront applicables à tout projet de supervision. Ces choix stratégiques, car ils vont grandement influencer la stratégie de la cellule *Availability* dans le cadre des projets de supervision, définiront, d'une part, le cadre dans lequel tout projet de supervision devra s'inscrire et, d'autre part, l'architecture à mettre en place pour rationaliser et optimiser les supervisions.

Ces choix stratégiques se traduiront donc par :

- ✓ La formulation de règles de supervision.
- ✓ Un choix d'architecture.
- ✓ La définition de profils de supervision.
- ✓ La standardisation des sévérités.

[15] Voir Audit page 26.

3.2 Règles de supervision

3.2.1 Définitions et formalisme

Comme nous l'avons vu, la méthodologie EMD[16] propose l'utilisation de règles de supervision qui vont fixer les limites et contraintes applicables à tout projet de supervision. Ces règles vont être dictées par l'environnement et les priorités du département informatique et par l'expérience du département ESM[17] en matière de supervision. Elles décrivent le cadre dans lequel le processus de supervision va s'inscrire et vont régir les droits et devoirs de toute personne ou département impliqué dans un projet de supervision.

Le formalisme préconisé par la méthodologie doit être légèrement modifié afin de mettre davantage en balance les justifications et les impacts de la règle. La formulation va se présenter comme suit :

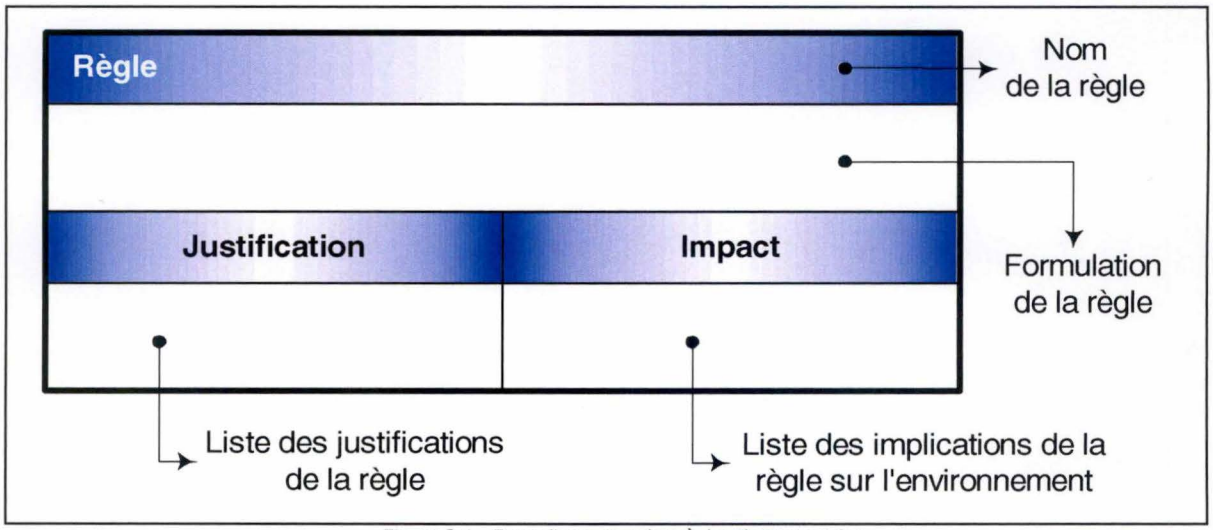


Figure 3-1 Formalisme pour les règles de supervision

Où : Nom décrit en un ou deux mots le sujet de la règle. Afin d'éviter toute confusion entre les règles, on veillera à l'unicité du nom pour toutes les règles.

Formulation est l'énoncé de la règle. Dans cette formulation, on veillera à souligner les mots clés afin de mettre en évidence ce qui est important dans la règle. De plus, pour mettre en évidence les idées principales de la règle, on en soulignera les mots ou les morceaux de phrases importants.

Justification dresse une liste des raisons qui ont incité sa rédaction. Celle-ci permet de réviser la règle en cas de changement de l'environnement.

Impact représente l'influence que peut avoir la règle sur l'environnement. Il permet également de détecter d'éventuels conflits ou contradictions entre les différentes règles.

[16] Voir Event Management Design (EMD) page 33.
[17] Voir Le département ESM : Enterprise Systems Management page 18.

3.2.2 Qualités d'une règle de supervision

Pour éviter des règles inutiles, longues ou incompréhensibles, EMD préconise les critères de qualité suivants :

- ✓ être nécessaires;
- ✓ avoir un impact sur l'organisation, les outils, ...;
- ✓ être formulées brièvement et simplement : une seule phrase est l'idéal, trois un maximum;
- ✓ être compréhensible par des personnes sans connaissance technique. Elles doivent donc éviter tout jargon technique.

Ces critères doivent bien évidemment être gardés afin, d'une part, de garder une qualité élevée de ces règles et, d'autre part, que leur lecture reste aisée pour tout lecteur non averti.

3.2.3 Classification des règles de supervision

Pour faciliter leur révision, dans le cas par exemple d'une modification de l'organisation, il est nécessaire de les classer. Cette classification va se baser sur les cinq grandes catégories de la même méthodologie EMD[18]. Pour rappel, ces catégories sont classées par domaine abordé, du plus général au plus ciblé :

- ✓ Entreprise.
- ✓ Organisation.
- ✓ Outils et plates-formes.
- ✓ Relations extérieures.
- ✓ Plates-formes cibles.

Pour ces cinq catégories de règles, voici les domaines abordés par chacune d'elles ainsi que quelques exemples.

Règles "Entreprise"

Les règles "Entreprise" décrivent la politique de supervision au sein de l'entreprise. Ces règles vont notamment régir les problématiques suivantes :

- ✓ Définition des responsabilités au sein de l'entreprise en matière de supervision.
- ✓ Le choix d'une interface utilisateur commune, l'intégration et le choix des outils de supervision et des protocoles à utiliser pour la remontée des alarmes.
- ✓ Le niveau d'automatisation du traitement des alarmes.
- ✓ Les informations obligatoires devant être contenues dans messages d'alarme.
- ✓ Le type de supervision à privilégier (pro-active, réactive ou intégrée).
- ✓ Les qualités requises des outils, des moniteurs, ...
- ✓ Définition de notions, de concepts, ...
- ✓ Nécessités ou besoins du département ESM.

[18] Voir Event Management Design (EMD) page 33.

Exemple 1 : Définition des responsabilités au sein de l'entreprise en matière de supervision.

Règle	**Choix des outils**
La sélection des outils de supervision est basée sur stratégie de supervision du département ESM, des standards utilisés et de l'architecture technique.	
Justification	**Impact**
➢ Fourniture d'un ensemble d'outils de supervision cohérent et consistant. ➢ Sélectionner rapidement des produits. ➢ Assurer la compatibilité entre les outils. ➢ Réduire la redondance des outils.	➢ ESM peut être amené à adopter des solutions techniques pouvant aller à l'encontre de la stratégie d'informatisation du département informatique.

Exemple 2 : Les qualités requises des outils, des moniteurs.

Règle	**Propriété des supervisions**
La supervision doit être réutilisable, indépendante de la plate-forme et avoir un cycle de vie de développement le plus court possible.	
Justification	**Impact**
➢ Minimiser les nouveaux développements. ➢ Avoir un processus de supervision uniforme à travers l'infrastructure informatique. ➢ Assurer la portabilité de la supervision en cas de changement de type de plate-forme. ➢ Réduire au maximum le cycle de vie des développements de supervision.	➢ Sélection de langages de développement universels. ➢ Réutilisation de l'ensemble des supervisions existantes doit être imposée.

Exemple 3 : Définition de notions, de concepts, ...

Règle	**Significatif Entreprise**
Pour être qualifiée de "Significatif Entreprise", une alarme doit signaler une réduction potentielle ou réelle dans le niveau de service pouvant amener une incapacité à assurer les qualités de services requises pour une application ou une plate-forme donnée.	
Justification	**Impact**
➢ Le processus de supervision doit, au niveau de l'entreprise, mettre l'accent sur les alarmes qui ont un impact réel sur le comportement des applications.	➢ Pour chaque alarme de ce type, il doit exister une procédure permettant de résoudre le problème.

Exemple 4 : Nécessités ou besoins du département ESM.

Règle	**ESM Implication**
Les conseils en matière de supervision donnés par le département ESM doivent être pris en considération le plus tôt possible dans le cycle de vie du développement des nouvelles applications.	
Justification	**Impact**
➢ Aptitude à concevoir des applications en tenant compte de la problématique de supervision. ➢ Rendre les applications plus faciles à superviser.	➢ ESM doit procurer des directives de développement pour les nouveaux développements. ➢ ESM doit être consulté à chaque étape du cycle de vie du développement.

Règles "Organisation"

Les règles "Organisation" régissent l'organisation ainsi que l'attribution des responsabilités pour tout projet de supervision. Ces règles vont notamment régir :

- ✓ La propriété, c'est-à-dire qui est le responsable des alarmes et de ses traitements.
- ✓ La localisation du traitement des alarmes : centralisé, distribué, ...
- ✓ La manière dont des sites informatiques distribués vont coopérer.
- ✓ Les conditions que doivent remplir toute supervision : ne pas dégrader les performances d'une plate-forme, être invisibles pour l'utilisateur...
- ✓ Les responsabilités et les propriétés dans l'organisation des projets de supervision.

Exemple 1 : La localisation du traitement des alarmes.

Règle	**Supervision centralisée**
La supervision, tant au niveau organisationnel que technique, doit être centralisée.	
Justification	**Impact**
➢ Assurer un contrôle de l'infrastructure de haut niveau. ➢ Optimiser les ressources techniques et humaines dans le domaine de la supervision.	➢ Utilisation d'outils de contrôle à distance pour des tâches de gestion.

Exemple 2 : Fixer les responsabilités et les propriétés dans l'organisation des projets de supervision.

Règle	**Propriété du processus**
La gestion du processus de développement pour la supervision est assurée par le département ESM qui en garantit la qualité et l'intégrité.	
Justification	**Impact**
➢ Empêcher des conflits de responsabilités ou des responsabilités non assumées.	➢ Le cycle de vie complet du projet de supervision doit être organisé et coordonné par le département ESM en collaboration avec les autres départements.

Règles "Outils et plates-formes"

Les règles "Outils et plates-formes" régissent l'utilisation des outils et des plates-formes dans le cadre des supervisions. Ces règles vont notamment aborder les problématiques suivantes :

- ✓ Localisation du filtrage des alarmes non "Significatif Entreprise".
- ✓ Synchronisation du statut des alarmes parmi tous les outils de traitement et de visualisation des alarmes.
- ✓ Définition du rôle des différents outils de supervision.
- ✓ Définition des relations entre les outils et les alarmes.

Exemple 1 : Définition des relations entre les outils et les alarmes.

Règle	**Filtrage**
Le filtrage des alarmes non "Significatif Entreprise" doit être effectué le plus tôt possible dans le cycle de vie de l'alarme. La localisation optimale pour le filtrage est à la source.	
Justification	**Impact**
➢ Réduire l'impact des alarmes non "Significatif Entreprise" sur le réseau, les ressources et les outils de supervision. ➢ Réduire la pollution d'alarmes non significatives sur les consoles. ➢ Optimiser l'identification des alarmes critiques et les temps de réaction à ces alarmes. ➢ Rationaliser et optimiser les règles de corrélation.	➢ Stocker tous les alarmes à la source pourrait être nécessaire pour permettre un audit complet des problèmes. ➢ Disposer d'outils intermédiaires de gestion pourrait être nécessaire dans des environnements où le filtrage des alarmes n'est pas possible.

Exemple 2 : Définition du rôle des différents outils de supervision.

Règle	Historique
L'historique des problèmes n'est pas maintenu au niveau des consoles *T/EC*.	
Justification	**Impact**
➢ Les alarmes trop anciennes ou clôturées sont considérées comme non "Significatif Entreprise".	➢ Si un historique est nécessaire, un ticket doit être envoyé vers l'outil du *Help Desk* afin d'y être archivé.

Règles "Relations externes "

Les règles "Relations externes" décrivent les relations entre le processus de supervision et les autres processus. Ces relations seront définies notamment par :

- ✓ L'utilisation par les outils de supervision des standards définis dans d'autres processus du département informatique.
- ✓ La synchronisation de la mise en production d'une supervision avec d'autres processus.
- ✓ La définition de degrés de sévérité des alarmes ainsi que leurs critères d'utilisation.

Exemple 1 : L'utilisation par les outils de supervision des standards définis dans d'autres processus du département informatique.

Règle	Standard *T/EC*
Les standards de sévérité des *Help Desk* doivent être utilisés dans les consoles *T/EC*.	
Justification	**Impact**
➢ Assurer une gestion efficace par le support de niveau 1 des ressources supervisées.	➢ ESM doit fournir une liste des degrés de sévérité aux responsables de la source des alarmes (application ou plate-forme). ➢ ESM doit définir des standards dans les degrés de sévérité utilisés pour les alarmes.

Exemple 2 : La synchronisation de la mise en production d'une supervision avec d'autres processus.

Règle	Planning
La mise en production d'une solution de supervision doit être synchronisée dans le temps avec la planification du département responsable de la gestion des versions des plates-formes techniques.	
Justification	**Impact**
➢ Un planning d'installation des versions est fixé pour un an et ne peut être retardé ou changé.	➢ Le planning de déploiement des solutions de supervision doit tenir compte de ce planning.

Règles "Plates-formes cibles"

Les règles "Plates-formes cibles" décrivent les standards que devront respecter les outils et moniteurs au niveau des ressources. Cela s'exprime, par exemple, par :

- ✓ La définition des standards des plates-formes et des applications.
- ✓ La localisation de la supervision d'une ressource : au niveau de la ressource, centralisée, ...
- ✓ La définition de contraintes techniques pour la supervision.
- ✓ La définition des standards de supervision auxquels doivent répondre les applications à superviser.

Exemple 1 : Définition des standards des plates-formes et des applications.

Règle	**Standard de supervision**
Les applications doivent se conformer à un minimum de standards de supervision.	
Justification	**Impact**
➢ Assurer un seuil minimum de supervision. ➢ Réduire le coût du support. ➢ Faciliter le développement de la supervision. ➢ Permettre une meilleure intégration des applications d'une même plate-forme.	➢ Des standards de supervision doivent être définis. ➢ Les standards de supervision doivent être acceptés par les équipes de développement. ➢ Les standards de supervision influencent le choix des outils de supervision.

Exemple 2 : Définition des contraintes techniques pour la supervision.

Règle	**Intervalle**
Le temps minimum imposé entre deux exécutions du même moniteur de supervision est fixé à cinq minutes.	
Justification	**Impact**
➢ Eviter une surcharge de la machine.	➢ Un délai de cinq minutes peut se produire entre le début d'un problème et la remontée d'une alarme vers le serveur d'alarmes.

3.3 Architecture de supervision

3.3.1 Justification

Comme nous l'avons vu précédemment dans l'audit[19], de par la manière dont ils ont été définis, les profils de supervision posent d'énormes problèmes à la cellule *Availability*. Afin de résoudre ces problèmes, il faut revoir la manière dont sont supervisées les plates-formes techniques et les applications. A cette fin, une architecture par couche de supervision sera favorisée.

En effet, une telle architecture permet de rendre les supervisions des différents composants d'une même plate-forme indépendantes les unes des autres. Ainsi, par exemple, un changement de hardware ou de système d'exploitation n'entraînera pas de changement dans la supervision des applications et *vice versa*.

Cette indépendance des supervisions permettra une meilleure souplesse face aux changements tout en diminuant grandement le nombre de profils de supervision pour un même composant. En fait, d'un seul profil par composant et par plate-forme, on passera à un profil par composant et par type de plate-forme.

3.3.2 Couches de supervision

Les couches de supervision sont déterminées comme suit :

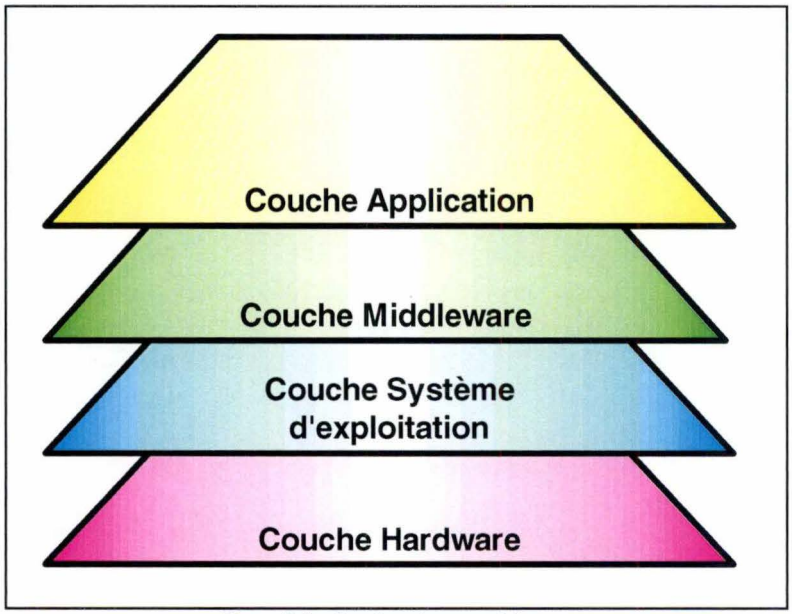


Figure 3-2 Couches de supervision

[19] Voir Audit page 26.

Voici quelques exemples de ce que recouvre chaque couche :

Couches	Exemples
Hardware	Machines *Compaq*, *Olivetti*, *OS/390*, *AS/400*...
Système d'exploitation	Serveur *NT*, *Unix*, *MVS*, *AIX*, *Solaris*...
Middleware	*MQSeries*, *Oracle*, *DB2*, *Exchange*...
Application	Applications propriétaires, *PC Banking*, *Web Banking*, *Call Center*...

3.3.3 Responsabilités de supervision

Afin d'éviter de superviser deux fois la même ressource ou le même composant à travers des profils de couches différentes, il convient de fixer les responsabilités ou les frontières attribuées à chaque couche en matière de supervision.

Par couche, ces responsabilités sont fixées comme suit :

Application

Les ressources propres à l'application telles que :

- ✓ les journaux applicatifs,
- ✓ les services, procédures propres à l'application,
- ✓ les connexions vers des bases de données,
- ✓ les connexions logiques avec d'autres applications,
- ✓ les composants internes de l'application : files d'attente, zones tampons, ...
- ✓ les systèmes de fichier spécifiques.

Middelware

Les ressources propres au middleware telles que :

- ✓ les journaux applicatifs,
- ✓ les services, procédures propres au middleware,
- ✓ les connexions vers des bases de données,
- ✓ les composants internes du middleware : files d'attente, zones tampons, ...

Système d'exploitation

Les ressources système telles que :

- ✓ le journal système,
- ✓ les services, procédures propres du système d'exploitation,
- ✓ la détection des virus,
- ✓ les disques logiques,
- ✓ la disponibilité des systèmes de fichiers,
- ✓ les archivages,
- ✓ les cartes mères, d'entrées et sorties, ...
- ✓ l'utilisation de la mémoire,
- ✓ l'utilisation du processeur.

Hardware

Les ressources physiques telles que :

- ✓ la mémoire et disques physiques,
- ✓ les cartes réseau,
- ✓ les contrôleurs,
- ✓ les portes physiques,
- ✓ les systèmes de tolérance de pannes : dédoublement des disques, système de protection contre les pannes de courant, ...

3.4 Profils de supervision

3.4.1 Types de supervision

On peut définir trois types de supervision :

- ✓ La supervision pro-active.
- ✓ La supervision réactive.
- ✓ La supervision intégrée.

La supervision pro-active réagit lors de changements d'état de fonctionnement du composant supervisé ou lors de l'atteinte de seuils prédéfinis dans l'utilisation de ressources de ce composant. Ces états et seuil sont vérifiés à intervalles réguliers par les moniteurs[20]. Pour la cellule *Availability*, ce genre de supervision est rapide et facile à mettre en œuvre.

Le changement d'état et le bon fonctionnement d'une ressource peuvent être :

- ✓ L'arrêt ou le démarrage d'un service système ou applicatif.
- ✓ L'établissement ou la fin d'une connexion logique ou physique.
- ✓ Le bon ou mauvais fonctionnement d'une carte réseau ou autre.
- ✓ La disponibilité de systèmes de fichiers, de données, ...
- ✓ La détection de virus, des atteintes à la sécurité, ...

L'atteinte de seuils peut être envisagée, par exemple, pour :

- ✓ L'utilisation de l'espace d'un disque logique.
- ✓ L'utilisation du processeur ou de la mémoire.
- ✓ Le changement de taille de fichiers, de files d'attente, ...
- ✓ Des indicateurs internes à la ressource (par exemple, occupation de zones tampons).

La supervision réactive réagit aux problèmes *a posteriori* en surveillant les journaux de systèmes ou d'applications. Des alarmes sont envoyées lorsque des messages répondant à des formats déterminés à l'avance y apparaissent. Ce genre de supervision est très difficile à mettre en œuvre car, dans la majorité des cas, les utilisateurs ont une grande méconnaissance des messages susceptibles de se retrouver dans de tels journaux. C'est d'autant plus vrai que l'on a affaire à des applications ou middleware achetés ou dont les développements ont été sous-traités. Dès lors, ce type de supervision est à éviter dans la mesure du possible.

Enfin, la supervision intégrée est assurée par des outils spécialement conçus pour un composant particulier. Il existe, par exemple, de telles supervisions pour *Oracle*, *NT*, *MQSeries*[21], ... Dans la plupart des cas, il s'agit d'un mélange de supervision pro-active et réactive auquel s'ajoute toute une série de tâches de contrôle et de configuration du composant. Par exemple, la supervision intégrée pour *MQSeries* comprend des moniteurs de supervision des services *MQSeries*, des moniteurs de contrôle de performance des files d'attente, des tâches d'arrêt ou de démarrage de connexions ainsi que de configuration de *MQSeries* lui-même. Bien quelle demande parfois certaines adaptations, ce type de supervision est à favoriser car elle offre une supervision complète pour un coût minimum.

Bien évidemment, chaque type de supervision ne permet pas de répondre à lui seul, à 100 %, aux besoins de supervision. C'est pourquoi, dans la plupart des cas, un profil de supervision sera un mélange de ces trois types. On veillera à construire toute supervision en utilisant un maximum des fonctionnalités offertes par un type de supervision avant de passer au suivant et ce, en respectant un ordre motivé par les caractéristiques décrites ci-dessus.

---

20 Voir Les outils de supervision page 19.

21 Middleware de messagerie électronique d'IBM. Les messages sont envoyés d'une plate-forme *MQseries* à une autre, ils sont d'abord stockés dans des files d'attentes puis envoyés vers l'autre plate-forme via des canaux de transmission propres au middleware appelés *Channel*.

Cet ordre est le suivant :

(1) Supervision intégrée car elle offre une solution complète.
(2) Supervision pro-active car elle est facile et rapide à implémenter.
(3) Supervision réactive car elle n'offre qu'une réaction *a posteriori* aux problèmes et est difficile à mettre en œuvre.

3.4.2 Supervision générique et spécifique

Afin de rationaliser le nombre de profils de supervision, il serait idéal de ne définir qu'un profil de supervision unique pour tous les composants d'un même type, par exemple, un profil de supervision pour les serveurs *NT*, un autre pour les serveurs *UNIX*, ... Cependant, les spécifications de supervision peuvent être différentes pour deux composants de même type suivant leur utilisation fonctionnelle. Ainsi, les seuils d'alerte d'utilisation du processeur pour un serveur de fichiers ne seront pas les mêmes que pour un serveur applicatif. Afin de pouvoir tout de même réduire au maximum le nombre de profils de supervision, la procédure suivante devra être utilisée :

(1) Définir des familles sur base des utilisations fonctionnelles possibles, des versions ou de particularités des divers composants. Afin de permettre une maintenance aisée des profils de supervision, 15 familles nous semble un maximum.
(2) Dégager les caractéristiques de supervision communes à toutes ces familles et définir un profil de supervision commun à toutes les familles. Cette supervision sera appelée supervision générique.
(3) Par famille, spécifier un profil de supervision propre appelé supervision spécifique.

Par type de composant, on aura un maximum de 16 profils[22]. Si les profils de supervision générique et spécifique peuvent théoriquement être construits sur base de n'importe quel type de supervision[23], une exception doit cependant être faite pour la supervision intégrée. En effet, les produits intégrés ne permettent pas d'être divisés en plusieurs parties. La spécification de la supervision utilisant de tels produits doit être unique. Cela implique qu'ils ne pourront être utilisés que dans des profils de supervision générique.

---

[22] 1 profil générique + 15 profils spécifiques.
[23] Voir Types de supervision page 52.

La supervision d'une plate-forme peut être représentée comme suit :

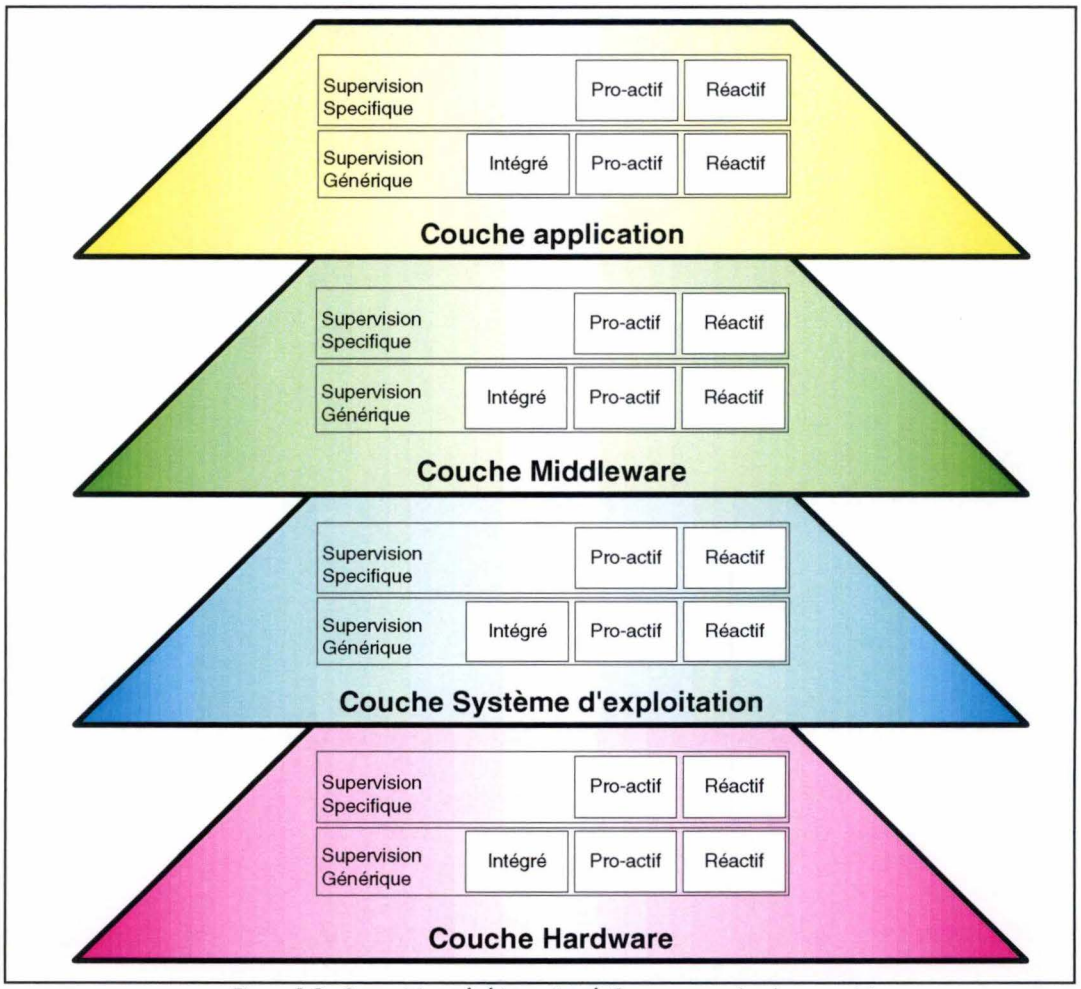


Figure 3-3 Supervision générique et spécifique par couche de supervision

3.4.3 Exemple

A titre d'exemple, prenons la supervision de la couche "Système d'exploitation" des serveurs *NT*. Pour ces serveurs, reprenons la procédure décrite précédemment[24] :

(1) Définition de familles de serveurs :

Les familles sont définies comme suit :

- ✓ Contrôleurs de domaine.
- ✓ Serveurs de fichiers.
- ✓ Serveurs d'impression.
- ✓ Serveurs d'application.
- ✓ Serveurs CD / DVD.
- ✓ Serveurs de base de données.
- ✓ Serveurs de courrier électronique.
- ✓ Serveurs WEB.
- ✓ Serveurs multifonctions type A.
- ✓ Serveurs multifonctions type B.
- ✓ Serveurs multifonctions type C.

Les serveurs multifonctions sont des serveurs regroupant deux ou plusieurs fonctions. Exemple : un serveur avec les fonctions d'impression et de serveur de données.

[24] Voir Supervision générique et spécifique page 53.

(2) Définition de la supervision générique :

Le profil de supervision générique va être défini comme suit :

Supervision intégrée pour *NT* :

- ✓ Mémoire centrale : occupation et pagination.
- ✓ Carte réseau : taux d'erreur, vitesse de traitement de la carte, segmentation.
- ✓ Disque logique : temps d'accès, taux d'utilisation et rapidité du disque logique.
- ✓ Processeur : occupation, performances et dysfonctionnements logiques.

Supervision pro-active :

- ✓ Services : état des services universels (*NetLogon*, *EventLog*, *RPC*, *Server*, *SNMP*, ...), occupation et pagination.
- ✓ Disque logique : occupation.

Supervision réactive :

- ✓ Rien.

(3) Définition des supervisions spécifiques pour chaque famille :

A titre d'exemple, spécifions la supervision des serveurs "Contrôleur de domaine" :

Supervision pro-active :

- ✓ Services : état des services "*Director Replicator*" et "*Computer Browser*".

Supervision réactive :

- ✓ Rien.

3.5 Niveaux de supervision

3.5.1 Définition

Selon les besoins du client en matière de supervision, plusieurs niveaux de supervision peuvent lui être proposés. Ces niveaux sont basés sur la profondeur de la supervision, c'est-à-dire l'aptitude à superviser et gérer efficacement le système. La supervision sera d'autant plus complète que le niveau de supervision implémenté sera élevé. La figure suivante montre les différents niveaux de supervision.

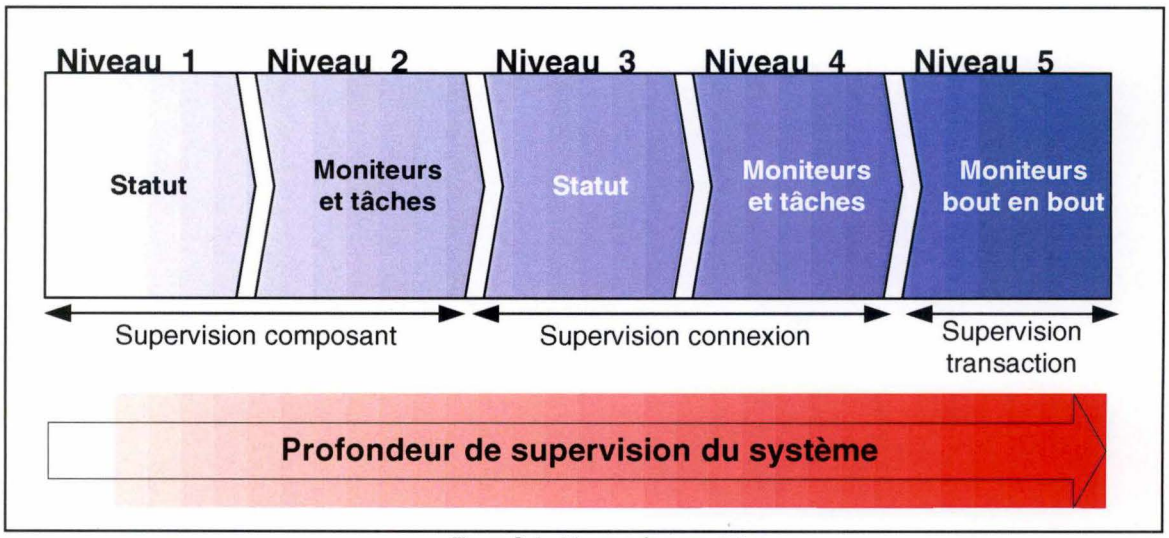


Figure 3-4 Niveaux de supervision

Avec, pour chaque niveau :

Niveau	Supervision
1	Supervision binaire, c'est-à-dire n'envoyant que des alarmes "En service" ou "Hors service", des ressources du système.
2	Supervision des composants du système avec des indicateurs plus élaborés et complétés par des tâches de contrôle ou de configuration permettant d'agir sur ces composants.
3	Supervision binaire des connexions entre les composants du système et d'autres systèmes.
4	Supervision fournissant des indicateurs plus élaborés sur l'état ou l'utilisation (volumes, temps d'attente, ...) des connexions.
5	Supervision de bout en bout des transactions utilisateurs mesurées d'un point de vue fonctionnel et performances.

On observe aisément que tout niveau de supervision est obligatoirement implémenté avec ses niveaux inférieurs. Par exemple, si le niveau 3 est choisi, les niveaux 1 et 2 devront être automatiquement implémentés.

A noter que, par transaction, nous désignons tous les traitements nécessaires (programmes, transfert des données, consultation ou mise à jour d'une base de données) que doit effectuer un SI pour accomplir l'une de ses fonctionnalité. Par exemple, la transaction de consultation du solde d'un compte en banque consiste à :

- ✓ Vérifier la syntaxe du code de client.
- ✓ Envoyer ce code via le réseau.
- ✓ Lire le solde.
- ✓ Recevoir et afficher le solde.

3.5.2 Exemple

Pour illustrer ces niveaux, reprenons l'exemple de *MQSeries* présent sur un serveur *NT*. Pour ce middleware, les cinq niveaux de supervision seraient définis comme suit :

Niveau	Supervision
1	Supervision du service *NT* de base (mqseries) de MQSeries. Des alarmes seront envoyées pour signaler que le service est démarré ou arrêté.
2	Niveau **1 +** Supervision du taux de remplissage de la file d'attente ABC *MQSeries*. Une alarme sera envoyée si plus de 1000 messages sont en attente dans cette file et une autre suivra lorsque tous les messages seront envoyés. Tâches d'arrêt et de redémarrage du service *MQSeries*.
3	Niveau **2 +** Supervision de l'état du *Channel*[25] XYZ. Une alarme sera envoyée lors de l'arrêt et du démarrage du *Channel*.
4	Niveau **3 +** Supervision du taux d'erreur sur le *Channel* XYZ. Une alarme sera envoyée si le nombre d'envois infructueux de messages sur ce *Channel* est supérieur à 3 %. Tâches d'arrêt et de redémarrage du *Channel* XYZ.
5	Niveau **4 +** Supervision de la disponibilité et des bonnes performances du chemin entre deux serveurs *MQSeries*. Envoi d'une transaction[26] sur le *Channel* XYZ et génération d'une alarme si le temps écoulé entre l'envoi de la transaction et la réception de la réponse est supérieur à 10ms.

[25] Connexion logique entre deux serveurs *MQSeries*.
[26] Voir Niveaux de supervision page 56.

3.6 Standardisation des sévérités

La sévérité d'une alarme indique la gravité de la panne signifiée par celle-ci. Les outils de supervision utilisent cinq niveaux ou degrés de sévérité : fatal, critique, mineur, avertissement et normal. Afin de faciliter l'évaluation de la gravité d'un problème en fonction du degré de la sévérité de l'alarme et ce, quel que soit le type de composant sur lequel ce problème apparaît, il convient de standardiser l'utilisation de ces degrés de sévérité. De plus, afin de localiser le plus rapidement sur les consoles *T/EC*[27] les alarmes indiquant un problème grave des autres alarmes, il convient d'associer uniformément une couleur à chaque degré de sévérité.

Standardisation des degrés de sévérité

Le degré de sévérité d'une alarme indique la gravité de la panne signifiée par celle-ci. Afin de facilité la détermination de la gravité d'un problème, il convient de standardiser comme suit, l'utilisation des différents degrés de sévérité en fonction de la gravité de la panne :

- ✓ Fatal — L'incident lié à l'alarme provoque une indisponibilité totale de l'application. Dans ce cas, une intervention immédiate est requise pour résoudre le problème.
- ✓ Critique — Une indisponibilité partielle et à court terme, totale, de l'application si aucune intervention n'est effectuée très rapidement pour résoudre le problème. Cette sévérité indique généralement que l'application fonctionne en mode dégradé et ne peut offrir toutes ses fonctionnalités ou ses performances habituelles.
- ✓ Mineure — Indique un problème de même sévérité que "critique" ou "fatal". La différence réside dans le fait que l'application est opérationnelle sur une plate-forme technique équipée d'un système particulier tel qu'un système en grappe, disque dédoublé ou connexions dédoublées, lui permettant de continuer de fonctionner. Ce type de problème ne nécessite pas une intervention urgente mais devra être résolu à court terme afin qu'une seconde panne ne provoque une indisponibilité réelle de l'application.
- ✓ Avertissement — Un problème minime n'entravant en rien la disponibilité de l'application est survenu. Ce genre d'alarme est utilisé pour apporter un complément d'information aux alarmes de sévérité supérieure ou pour indiquer une dégradation minime de l'état de fonctionnement de l'application. Cette dégradation pourrait, à moyen terme, engendrer des problèmes plus graves.
- ✓ Normal — Fin d'un incident de n'importe quelle sévérité. C'est le degré de sévérité qui sera notamment utilisé pour les alarmes de type *Clearing*[28].

[27] Voir Les outils de supervision page 19.
[28] Voir Event Management Design (EMD) page 33.

Standardisation des degrés de sévérité

Dans le même ordre d'idée, afin de faciliter aux équipes de support[29] la lecture de toutes les alarmes sur les consoles *T/EC*[30], il convient également d'uniformiser comme suit l'utilisation des couleurs pour les différents degrés de sévérité :

Couleurs	Degrés de sévérité
Noir	Fatal
Rouge	Critique
Orange	Mineur
Jaune	Avertissement
Vert	Normal

Nous avons fixé ces couleurs en fonction des couleurs disponibles sur les consoles *T/EC*, l'idée étant de répartir les couleurs en partant de la plus foncée (noir) au degré de sévérité le plus élevé pour terminer par la plus claire (vert) au degré le plus bas.

[29] Voir Support aux utilisateurs page 24.
[30] Voir Les outils de supervision page 19.

4 Méthodologie

4.1 Introduction

4.1.1 Découpe de la méthodologie

Sur base de ces choix stratégiques, nous pouvons maintenant bâtir une méthodologie qui va nous permettre de mener un projet de supervision de bout en bout tout en rencontrant tous les objectifs que nous nous sommes fixés[31]. A noter que les objectifs ① et ② ont déjà été atteints grâce aux profils de supervision et à l'architecture définis dans les choix stratégiques[32].

La méthodologie peut se résumer en une succession de différentes phases. Chaque phase est divisée en une ou plusieurs activités. Celles-ci peuvent être exécutées une ou plusieurs fois, de manière itérative ou simplement omises. Les activités peuvent générer des documents ou des développements qui seront utilisés par une ou plusieurs activités ultérieures faisant ou non partie de la même phase.

[31] Voir Conclusion du chapitre Audit page 26.
[32] Voir Choix Stratégiques page 41.

Une activité peut être :

- ✓ Une présentation : durant laquelle la cellule ESM va expliquer ou clarifier une problématique, une phase ou des documents utilisés dans la méthodologie.
- ✓ Une réunion : au cours de laquelle la cellule *Availability* rencontre le client ou d'autres départements afin de débattre de questions relatives au projet de supervision.
- ✓ Une séance de travail : durant laquelle la cellule *Availability*, le client ou tout autre participant au projet de supervision travaille, seul ou ensemble, à la réalisation de la supervision.

4.1.2 Plan général

La méthodologie se divise en cinq phases ayant chacune leurs objectifs bien particuliers :

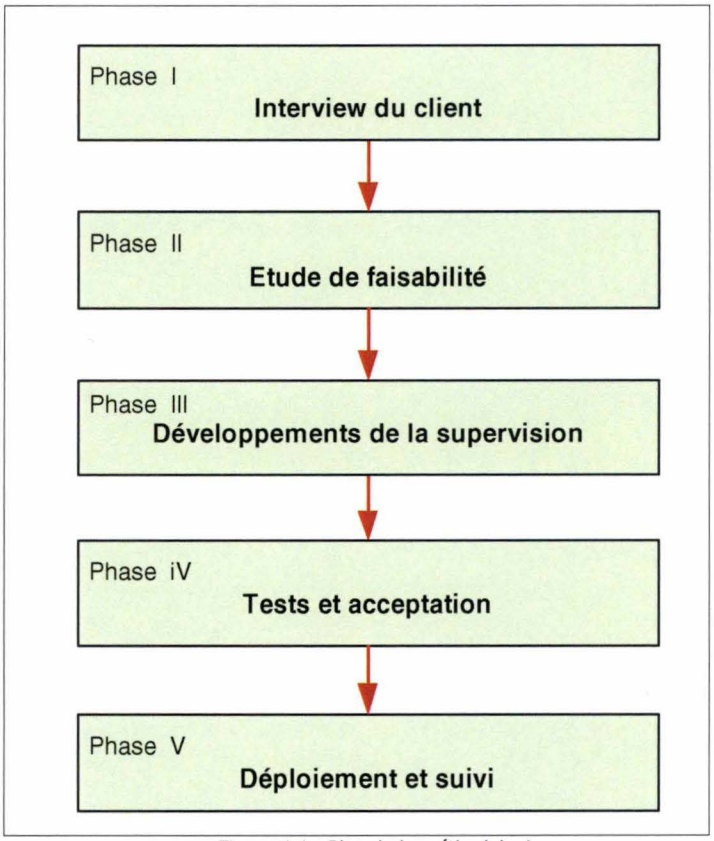


Figure 4-1 Plan de la méthodologie

L'objectif principal des différentes phases sont les suivants :

- Phase I Collecte d'informations liées au SI et des spécifications fonctionnelles auprès du client.
- Phase II Estimation des charges de travail, spécifications techniques et élaboration de la planification du projet.
- Phase III Développements des programmes liés au projet de supervision.
- Phase IV Tests de la solution et acceptation de cette solution par le client.
- Phase V Déploiement et suivi de qualité de la supervision.

4.1.3 Les acteurs

Lors des différentes phases du projet de supervision, différentes personnes ou acteurs seront susceptibles d'intervenir. Ces acteurs sont :

- ✓ Le représentant de la cellule *Availability*.
  Cette personne est généralement un chef de projet de la cellule. Il sera chargé de collecter toutes les informations du client, de réaliser l'étude de faisabilité et du suivi du projet de supervision. Il sera, pour le client, son point de contact dans le département ESM.

- ✓ Le chef de département.
  Il s'agit ici du chef du département duquel dépend le SI à superviser.

- ✓ Le chef de projet.
  Il s'agit une fois encore du chef de projet du SI à superviser. Sa participation est évidente car c'est la personne qui devra prendre toutes les décisions liées au projet.

- ✓ Correspondant technique.
  Il s'agit d'un membre du département du client ayant une connaissance approfondie (tant fonctionnelle que technique) du SI à superviser. Cette personne sera désignée par le chef de projet du SI et sera, pour la cellule *Availability*, le point de contact unique chez le client durant le projet.

- ✓ L'architecte du SI.
  Est la personne qui a mis en place l'architecture du SI. Cette personne connaît bien évidemment la topologie du SI mais également tous les produits (logiciels, matériel, middleware,...) du SI ainsi que les relations qui lient ces différents composants.

- ✓ Les départements techniques.
  Sont les département en charge du déploiement et de la maintenance des plates-formes techniques (système d'exploitation ou middleware).

- ✓ Les experts techniques.
  Sont les représentants des différents départements techniques responsables de l'installation, de la maintenance et du suivi d'un composant technique tel qu'un système d'exploitation, un middleware ou un réseau.

- ✓ Le représentant de la *Master Console*.
  Est le représentant de l'équipe de support *Master Console*. C'est le point de contact de la cellule Availability auprès de cette équipe.

En réalité, il n'est pas rare qu'un acteur remplisse plusieurs fonctions. Par exemple, le chef de projet peut également être l'architecte du SI et être, en plus, désigné comme correspondant technique pour le projet.

4.1.4 Les documents

Comme nous l'avons vu, une série de documents seront produits tout au long des différentes phases et activités. Afin d'utiliser un formalisme unique pour tous les projets, des modèles seront réalisés pour chaque document. Ces modèles de documents seront appelés ***formulaires***. Une fois complété, un formulaire est appelé ***document instancié***.

Les formulaires et documents instanciés auront tous la structure suivante :

Section I : Préface	Cette section contient trois parties : **Contexte et usage** Il s'agit ici de préciser le cadre d'utilisation du document, en d'autres mots, préciser que le document est utilisé dans le cadre de la méthodologie destinée aux projets de supervision. Il faut également décrire à quelle étape de la méthodologie le document est utilisé. Ceci n'a d'autre objectif que de restituer le document pour tout lecteur qui l'aborderait hors du contexte méthodologique. **Convention de nom** Cette partie n'est présente que pour un formulaire. Il s'agit de spécifier les conventions de nom que devra respecter tout document instancié tiré de ce formulaire. Cette section devra disparaître d'un document instancié. **Gestion du document** Enfin, il faut spécifier un point de contact auprès duquel le client peut s'adresser pour toute question relative au contenu du formulaire ou pour obtenir une aide supplémentaire pour le compléter.
Section II : Objectifs	Il s'agit ici de décrire brièvement les objectifs poursuivis par le document afin de permettre au client de se situer dans le cycle de la méthodologie mais également de permettre à un lecteur "non averti" de cerner rapidement l'objet du document.
Sections III, IV, ...	Les sections suivantes composent le corps proprement dit du document.

Figure 4-2 Structure d'un formulaire

Enfin, pour constituer une base de données documentaire, les documents doivent non seulement être sauvegardés à un endroit unique mais également être faciles à retrouver. De plus, si une découpe du projet en plusieurs phases est adoptée, plusieurs versions d'un même document pourront coexister. Pour répondre à ces besoins, il faut prévoir :

1. Une arborescence structurée de répertoires dans lesquels seront sauvegardés les documents.
2. Une convention de nom pour les formulaires, les documents techniques et les documents instanciés.

L'arborescence suivante est proposée.

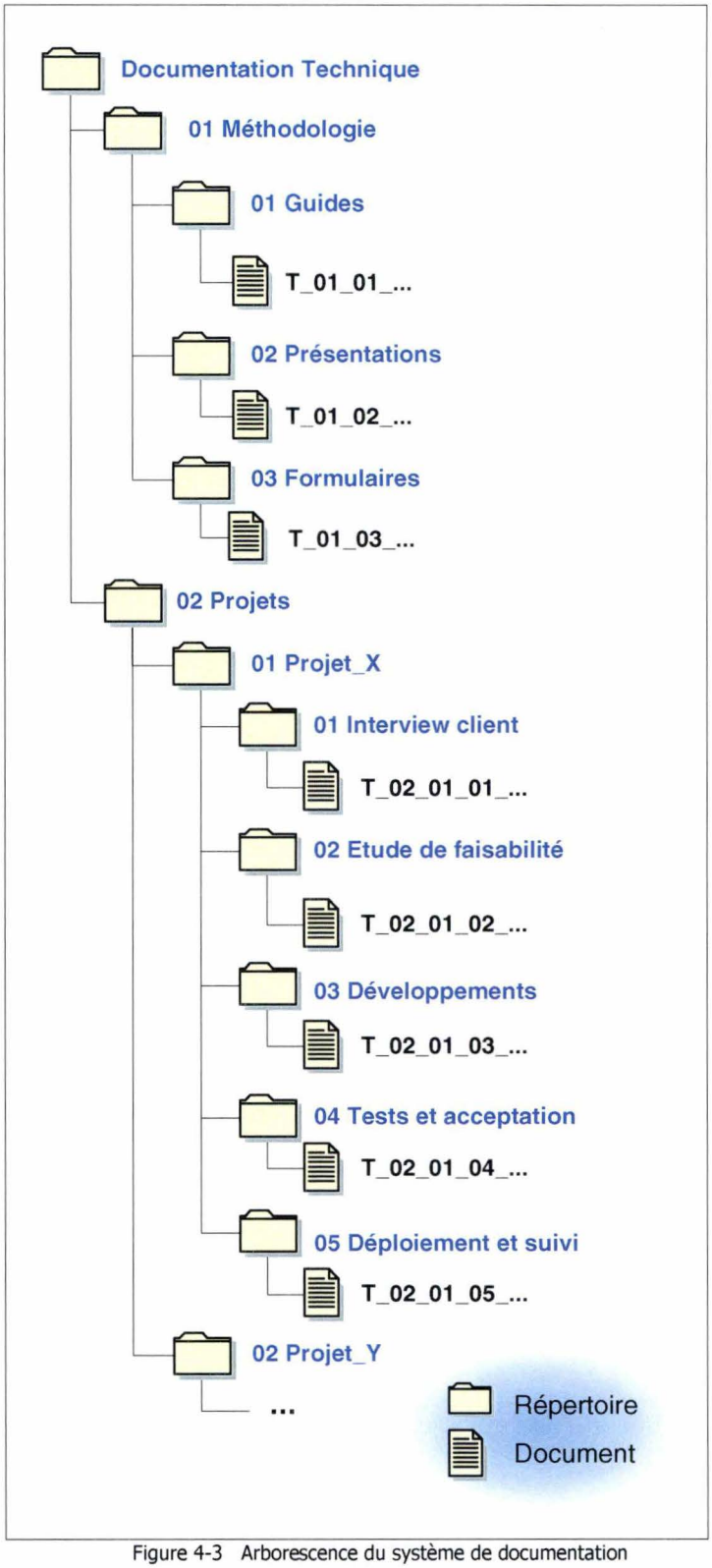


Figure 4-3 Arborescence du système de documentation

Tous les documents de base de la méthodologie sont regroupés dans le répertoire dédié à la méthodologie :

- ✓ Le guide de l'utilisateur et le catalogue dans le répertoire nommé ***Guides***.
- ✓ Les présentations dans le répertoire ***Présentations***.
- ✓ Les différents formulaires utilisés tout au long du projet dans le répertoire ***Formulaires***.

Les documents instanciés sont regroupés dans le répertoire ***Projets***. Il existe un répertoire par projet de supervision. Chacun de ces répertoires est subdivisé en sous-répertoires à raison d'un sous-répertoire par phase de la méthodologie. Ainsi, tous ces sous-répertoires contiennent tous les documents instanciés de chaque phase pour le projet concerné.

Afin de pouvoir facilement classer et retrouver les documents dans les différents répertoires et sous-répertoires, ceux-ci portent tous un numéro. Afin de déterminer facilement le répertoire où se trouve ou doit être sauvegardé un document, celui-ci suivra la convention de nom suivante :

Pour un formulaire :

T_{Chemin_Répertoire}_Fomulaire _{Contenu_Document}_v{Version}

Pour un document instancié :

T_{Chemin_Répertoire}_{Contenu_Document}_{Nom_de_Projet}_v{Version}

Où :	T	Préfixe pour la documentation technique.
	Chemin_Répertoire	Numéros des répertoires et sous-répertoires successifs.
	Contenu_Document	Type du document. Il s'agit d'un acronyme décrivant le type d'information contenue dans le document. Par exemple DSI pour "Description du Système d'Information".
	Nom de projet	Nom du projet concerné.
	Version	Version du document.

Exemple :

Soit la version *1.0* d'un formulaire de description logique du SI (DLSI) utilisé durant la phase I. Ce formulaire portera le nom *T_01_03_Formulaire_DLSI_v1.0*

La version *1.0* du document instancié pour le projet n°4 XYZ de ce formulaire portera, quant à lui, le nom *T_02_04_01_DLSI_YYZ_v1.0*

4.1.5 Le guide de l'utilisateur

Le guide de l'utilisateur est un document qui sera utilisé tout au long du projet de supervision et sera remis au client dès le début du projet. Il explique dans le détail, activité par activité, la méthodologie qui sera utilisée pour le projet. Il détaillera également l'architecture, les règles de supervision, la structure des profils, les niveaux de supervision et le contenu des documents utilisés lors des différentes activités. Enfin, ce guide énoncera les rôles de chaque participant dans le cadre du projet. Dans le cadre du système de documentation décrit plus haut, la version *1.0* du guide de l'utilisateur portera le nom *T_01_01_GuideUtilisateur_v1.0*.

4.1.6 Le catalogue

Le catalogue est un document reprenant l'inventaire complet des profils de supervision existants de composants techniques. Dans ce catalogue, on retrouvera les spécifications de toutes les supervisions des trois couches de supervision inférieures, à savoir les couches hardware, système d'exploitation et middleware. Ces profils seront décrits par leurs spécifications, le type et la version du composant technique auquel il s'applique ainsi que la

zone sécurité concernée (Intranet, zone sécurisée[33], extérieur...). Cette dernière information est importante car, pour des raisons évidentes de sécurité, l'utilisation des outils de supervision dans une zone sécurisée est très limitée.

Ces profils sont obligatoires et sont systématiquement installés sur toutes les instances des composants des profils. Dans le cadre du système de documentation, le nom de la version *1.0* du catalogue sera *T_01_01_Catalogue_v1.0*.

[33] La zone sécurisée d'un site informatique, appelée également zone démilitarisée ou DMZ, est la partie du réseau informatique de l'entreprise accessible par les réseaux publics (Internet par exemple). C'est pourquoi les plates-formes et le réseau de cette zone sont soumis à une sécurité très stricte.

4.2 Phase I : Interview du client

4.2.1 Objectifs

Il s'agit de la phase la plus longue et la plus importante de la méthodologie. En effet, comme nous l'avons vu, la cellule *Availability* rencontre de grandes difficultés pour obtenir de la part du client toutes les informations nécessaires pour la supervision. Cette phase va donc permettre de collecter de manière structurée et complète, toutes les informations (description fonctionnelle, technique, spécifications...) nécessaires pour mener à bien le projet de supervision. Les trois grands objectifs de la phase I sont les suivants :

- ✓ Expliquer au client la problématique de la supervision.
- ✓ Dresser un descriptif complet de l'application à superviser. Ceci comprend notamment les responsabilités, son caractère critique, les heures de disponibilité requises, les dates d'échéance, ...
- ✓ Collecter toutes les informations nécessaires pour être à même de concevoir une supervision appropriée.

4.2.2 Plan de la phase

La phase I, dont nous allons décrire les différentes activités et les documents utilisés tout au long de ce chapitre, se décompose comme suit :

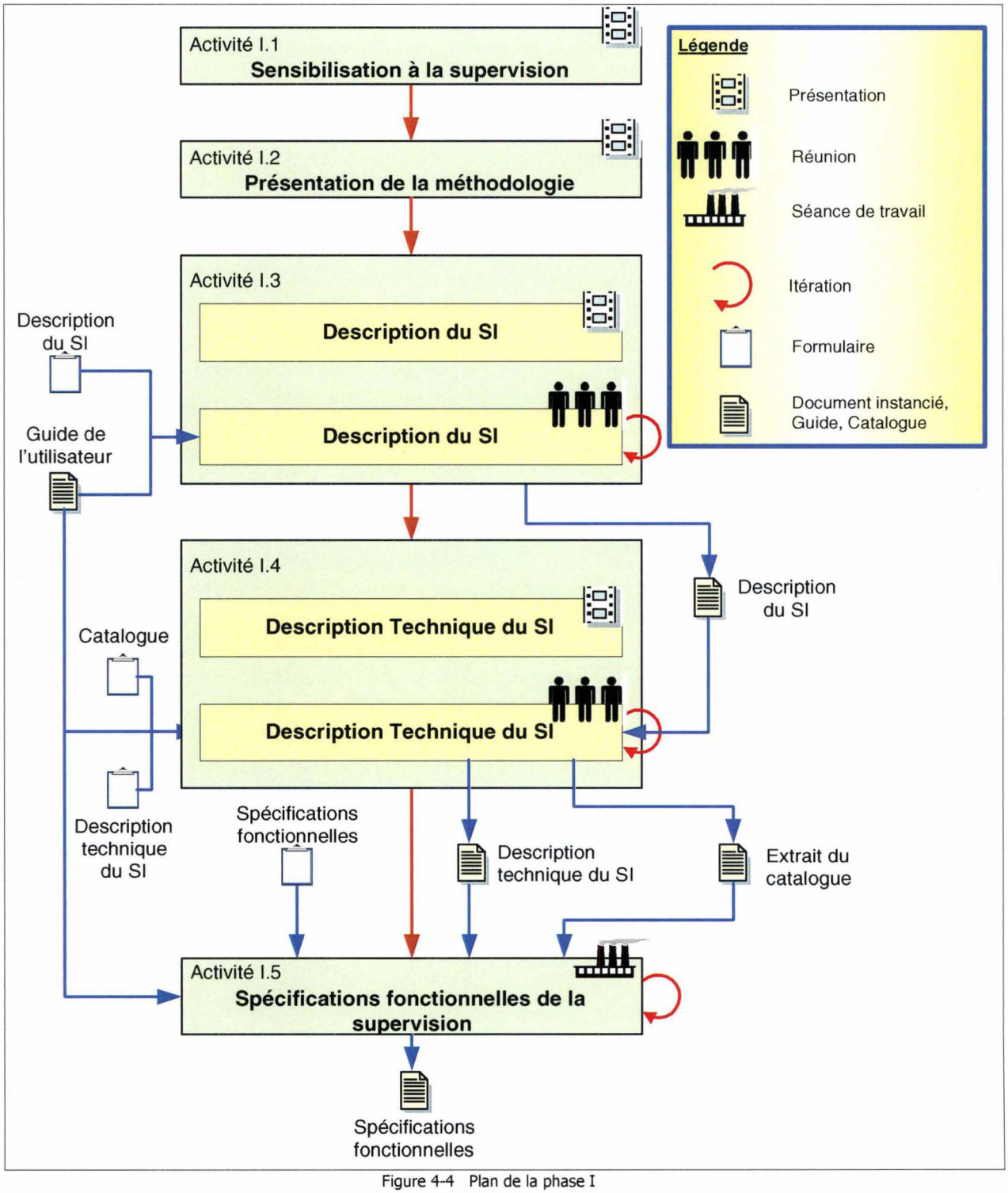


Figure 4-4 Plan de la phase I

4.2.3 Activité I.1 : Sensibilisation à la supervision

4.2.3.1 Objectifs

Cette activité, de type présentation, a comme objectifs de :

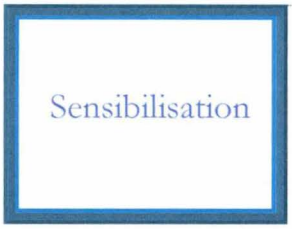
- ✓ Sensibiliser le client à la problématique de la supervision et à l'intérêt de prendre en compte cette problématique le plus rapidement possible dans le cycle de développement d'un nouveau projet.
- ✓ Expliquer comment la problématique de supervision a été abordée au sein du département informatique.

4.2.3.2 Contenu de la présentation

La présentation abordera les sujets suivants :

Présentation du département ESM	Dans la majorité des cas, pour le client, il s'agit de la première collaboration avec le département ESM. Il peut donc être intéressant de décrire l'organisation interne ainsi que les responsabilités au sein du département et plus particulièrement de la cellule *Availability*.
Définition de la supervision	Pour bon nombre d'informaticiens, la supervision se limite à la détection de pannes. C'est pourquoi il peut être nécessaire d'expliquer qu'elle ne se limite pas seulement à la détection des pannes mais également à l'acheminement et au suivi des alarmes. Il faut donc sensibiliser le client sur les informations qui lui seront demandées tout au long du projet pour couvrir tous ces aspects[34].
Les outils de supervision	Une description des outils de supervision utilisés et de leur rôle dans la supervision est bien évidemment nécessaire. Cette explication va permettre de fixer une partie du vocabulaire qui sera utilisé tout au long du projet de supervision.
L'architecture de supervision	Une explication de l'architecture de supervision en couches, implémentée par le département ESM, est également nécessaire. Elle va permettre au client de comprendre pourquoi les spécifications de supervision se limitent aux seuls composants dont il est responsable.

[34] Type de support possible, description des procédures de résolution des problèmes...



Pour conclure, il peut être utile de sensibiliser le client sur l'intérêt de superviser des applications et de prendre en compte la problématique de la supervision le plus tôt possible dans le cycle de vie d'un projet. En fait, le respect de quelques règles simples permet la mise en œuvre plus aisée d'une supervision. C'est d'autant plus vrai que le niveau de supervision désiré est élevé.

4.2.3.3 Participants

Puisque cette présentation n'aborde pas le projet de supervision en lui-même mais plutôt la problématique de la supervision de manière très générale, la présence de techniciens n'est pas nécessaire. De plus, la dernière partie de la présentation est principalement dévolue à une sensibilisation à la problématique de la supervision des responsables de département et des chefs de projet. Pour ces raisons, l'assistance requise à cette présentation peut se limiter aux personnes suivantes :

- ✓ Un membre de la cellule *Availability* pour assurer la présentation.
- ✓ Le chef de projet de l'application à superviser.
- ✓ Le chef du département duquel dépend le chef de projet.

La présence du chef du département n'est bien évidemment pas requise pour chaque projet de supervision mais au moins une fois. Il peut en effet être intéressant que celui-ci assiste à cette présentation pour, d'une part, se rendre compte de l'organisation liée au support aux utilisateurs et, d'autre part, éventuellement influencer les développements à venir afin de mieux tenir compte de la problématique de la supervision.

Cette présentation doit être considérée comme obligatoire car elle introduit non seulement toutes les notions et le vocabulaire nécessaires au bon dialogue entre le client et la cellule *Availability* mais explique également toute l'organisation mise en place pour le support. Elle peut bien sûr être omise si toutes les personnes concernées l'ont déjà suivie.

4.2.3.4 Résultats

Aucun document n'est produit lors de cette activité.

4.2.4 Activité I.2 : Présentation de la méthodologie

4.2.4.1 Objectifs

Comme son nom l'indique, cette activité consiste en une présentation de la méthode de travail qui sera utilisée pour la réalisation de la supervision. Elle a bien sûr comme principal objectif de décrire les différentes phases du projet et va sensibiliser le client sur différents aspects moins évidents du projet, comme :

- ✓ La charge de travail que représente un projet de supervision.
- ✓ Donner un aperçu des besoins en informations, en ressources informatiques et humaines qui seront nécessaires pour la réalisation du projet.
- ✓ Définir le rôle de chacun dans les différentes activités du projet.

4.2.4.2 Contenu de la présentation

Le contenu se présentera donc comme suit :

Positionnement de l'activité	Le positionnement de l'activité consiste à restituer l'activité en cours dans tout le cycle du projet.
Résumé des phases	Le résumé des phases va présenter dans les grandes lignes les quatre phases qui composent un projet de supervision.
Présentation des activités	La présentation des activités consiste, phase par phase, à détailler toutes les activités qui les composent. Cela va permettre au client de se faire une idée sur le temps et la charge de travail qu'il va devoir consacrer pour la collecte des informations et pour le projet proprement dit. Par activité, nous décrirons les informations suivantes : ✓ Les documents qui seront fournis par la cellule *Availability* au début de l'activité. ✓ Les informations nécessaires que le client est amené à fournir pour mener à bien l'activité. ✓ Les personnes impliquées dans l'activité. ✓ Les résultats générés par l'activité (documents, programmes...).
Activité suivante	Enfin, pour conclure, l'activité suivante va donner un aperçu de la prochaine activité afin de donner au client un aperçu des informations qui lui seront demandées, des tâches à effectuer ou des personnes à convoquer pour la prochaine activité.

4.2.4.3 Participants

De même que pour la présentation précédente, cette présentation n'aborde aucun côté technique du projet de supervision mais son organisation. L'assistance requise pour cette présentation peut donc se limiter aux personnes suivantes :

- ✓ Un membre de la cellule *Availability* pour assurer la présentation.
- ✓ Le chef de projet de l'application à superviser.

Cette présentation doit également être considérée comme obligatoire, c'est-à-dire imposée au client, car elle décrit tout le cycle que va suivre le projet. Elle va permettre au client de se faire une première idée des coûts tant humains que matériels et des délais que nécessitera le projet. Elle peut également être omise si toutes les personnes concernées l'ont déjà suivi ou travaillent dans le domaine de la supervision.

4.2.4.4 Résultats

A la suite de la présentation, le représentant de la cellule *Availability* va demander au client de désigner au sein de son département un correspondant technique. Il va également lui remettre une copie du guide de l'utilisateur[35] de la méthodologie.

4.2.5 Activité I.3 : Description du système d'information

4.2.5.1 Objectifs

L'unique objectif de cette activité est de collecter un maximum d'informations sur l'organisation du système d'information à superviser et d'expliquer au client pourquoi ces informations sont nécessaires et importantes dans le cadre du projet. Pour ce faire, nous allons diviser cette activité en deux parties :

- ✓ Une présentation au cours de laquelle seront expliqués l'objectif de l'activité I.3 et, point par point, le contenu du formulaire utilisé pour cette activité.
- ✓ Une réunion durant laquelle le client, avec l'aide d'un représentant de la cellule *Availability*, complètera le formulaire. Dans la majorité des cas, le client ne dispose pas de la totalité des informations nécessaires. Il s'agira surtout d'une série de réunions et de séances de travail pour le client destinées à la collecte des informations nécessaires.

4.2.5.2 Contenu de la présentation

La présentation se déroulera comme suit :

Positionnement de l'activité	Le positionnement de l'activité consiste à restituer l'activité en cours dans tout le cycle du projet.
Résumé de l'activité	Le résumé de l'activité va présenter ce qui sera réalisé durant l'activité, les résultats qui seront produits ainsi que les personnes requises pour la deuxième partie de l'activité.
Explication du formulaire	L'explication du formulaire consiste à expliquer dans le détail le contenu du formulaire qui sera utilisé lors de la deuxième partie de l'activité. Il faut non seulement énumérer les informations dont la cellule *Availability* a besoin mais également expliquer pourquoi ces informations sont nécessaires.

[35] Voir Le guide de l'utilisateur page 66.

Enfin, pour conclure, l'activité suivante va donner un aperçu de la prochaine activité afin de donner au client un aperçu des informations qui lui seront demandées, des tâches à effectuer ou des personnes à convoquer pour la prochaine activité.

4.2.5.3 Documents utilisés

Les documents utilisés pour la partie réunion de cette activité sont les suivants :

- ✓ Le guide de l'utilisateur dans lequel le client trouvera les réponses à ses éventuelles interrogations au sujet des activités de la méthodologie et de leurs formulaires.
- ✓ Le formulaire, destiné à la collecte des informations relatives à la description du SI utilisé dans la deuxième partie de l'activité, peut également être utilisé comme support lors de la présentation.

4.2.5.4 Formulaire de description du SI

Le formulaire utilisé au cours de la deuxième partie de l'activité est destiné à collecter un maximum d'informations sur le SI, notamment son organisation, son planning, les volumes traités, ... Toutes ces informations seront utilisées afin de connaître le SI, de se faire une idée de son caractère critique et de son importance pour l'entreprise et d'évaluer le préjudice subi par celle-ci en cas d'indisponibilité. Nous allons décrire dans le détail toutes les questions qui seront posées dans ce formulaire et expliquer en quoi ces informations sont nécessaires dans le cadre du projet de supervision. Certaines informations sont probablement connues de la cellule *Availability* soit parce qu'elles se trouvent dans le référentiel des applications[36], soit parce qu'il s'agit d'un SI qu'elle connaît. Afin de réduire le travail du client, la cellule Availability doit compléter le formulaire avec les informations déjà en sa possession et ce, avant de le remettre au client.

Les questions du formulaire vont s'orienter autour de six axes :

Organisation liée au SI

Les informations nécessaires concernant l'organisation du SI sont de deux types : des informations générales et les responsabilités.

Les informations générales vont permettre de constituer une fiche descriptive du système d'information. Cela comprend :

- ✓ Son nom
- ✓ Son code
  Va déjà permettre de savoir si l'application a suivi ou pas les cycles de développement standards en vigueur au sein du département informatique. En effet, seules les applications suivant ces standards en sont munis.
- ✓ Sa *Business Line*[37]
  Va permettre de situer l'application dans l'un des métiers de la banque.
- ✓ Son domaine :
  Va permettre de situer l'application dans les grands projets du département informatique.

[36] Ce référentiel est uniquement utilisé pour les applications.
[37] Voir Organisation de FORTIS Banque page 14.

- ✓ Sa date de mise en production
- ✓ Son mode de développement
  Est-ce une application développée au sein du département informatique ou par une société externe ou encore, est-ce un logiciel acheté ? Ces informations sont importantes car elles vont permettre de se faire une première idée des difficultés à venir. En effet, le dialogue avec les sociétés externes est toujours plus difficile et plus long qu'avec des services internes. En ce qui concerne les applications achetées, les possibilités de supervision sont très limitées car dans la plupart des cas, il n'est pas possible de superviser des composants internes à l'application. Enfin, les alarmes générées par de telles applications ne sont pas toujours des mieux documentées.

Les informations relatives aux responsabilités sont nécessaires pour permettre à la cellule *Availability* d'obtenir en une fois la liste de toutes les personnes susceptibles d'intervenir à un moment ou un autre dans le projet de supervision. La liste des responsables se compose comme suit :

- ✓ Le chef de projet.
- ✓ Le responsable des développements.
- ✓ Le responsable d'intégration. Celui-ci est responsable de l'intégration du SI avec les autres SI.
- ✓ Le responsable de l'organisation du projet.
- ✓ Le responsable de l'architecture.
- ✓ Et toute autre personne ayant joué un rôle important dans le cycle de développement du SI.

Planification

La majorité des projets de supervision soumis à la cellule *Availability* concernent des applications existantes. Dans ce cas, il peut être intéressant de connaître un peu l'historique des mises en production du projet afin de tirer certaines conclusions. Pour cela, trois dates de mise en production sont révélatrices :

- ✓ La date de la première version. Cette date va permettre de se faire une idée sur l'âge de l'application ainsi que la vétusté des technologies utilisées.
- ✓ La date de la dernière et de la prochaine version. Si ces deux dates sont proches dans le temps, cela peut signifier deux choses :
  - L'application n'est pas stable. Cette instabilité peut, par exemple, signifier soit que l'application est en phase de modernisation[38], soit que les utilisateurs modifient continuellement leurs exigences. Quelle qu'en soit la raison, cela va entraîner une même instabilité sur la supervision. Dans ce cas, si cela est possible, il conviendra de proposer de postposer la supervision d'une telle application jusqu'à sa stabilisation.
  - L'application connaît beaucoup de problèmes techniques ou applicatifs. Dans ce cas, une supervision pour aider au suivi du SI est probablement urgente. De même, il conviendra sans doute d'augmenter la priorité du projet.

Pour les projets concernant de nouvelles applications, il est nécessaire de connaître toutes les échéances du cycle de développement de l'application et ce pour trois raisons :

[38] La date de la première mise en production peut nous donner une indication à ce sujet.

- ✓ Pouvoir faire coïncider les différentes étapes du projet de supervision avec celles du développement de l'application.
- ✓ Si une nouvelle infrastructure doit être mise en place, ces dates vont permettre de connaître les échéances de mise à disponibilité de cette infrastructure et aider ainsi à la planification des tests de la supervision.
- ✓ Dans la plupart des cas, on demande que la supervision soit prête en même temps que l'application. Les échéances du SI supervisé vont donc devenir celles du projet de supervision.

Pour de tels projets, les dates intéressantes sont :

- ✓ Fin des spécifications.
- ✓ Fin du design de l'application.
- ✓ Fin des développements.
- ✓ Début et fin de la phase d'acceptation.
- ✓ Début et fin du déploiement.
- ✓ Mise en production.

Description du SI

La description du SI va permettre de se faire une idée plus précise des fonctionnalités de l'application et des éventuelles relations avec d'autres SI. Les informations demandées sont les suivantes :

- ✓ Objectifs du SI
  Description en quelques mots des objectifs poursuivis par le système d'information.
- ✓ Grandes fonctionnalités
  Enoncé dans les grandes lignes des principales fonctionnalités offertes aux utilisateurs par le SI. Ces fonctionnalités sont intéressantes à connaître car elles peuvent être considérées comme un composant à part entière du SI. Un tel composant peut être disponible ou non. En fait, la bonne santé d'un SI dépend tout autant de la disponibilité de ses composants techniques que de ses fonctionnalités.
- ✓ Relations et dépendances avec d'autres SI
  Cette question est très importante car si de telles relations ne sont pas soumises à la supervision, un SI risque d'être considéré comme disponible alors qu'en réalité il ne l'est pas. On peut citer en exemple un SI dépendant, pour ses données, d'un autre SI non soumis à la supervision. Si ce dernier est indisponible, alors le SI l'est plus que certainement aussi sans qu'aucune alarme ne soit visible. Cette question qui, à première vue, peut paraître secondaire, va donc permettre d'éviter des oublis dans la supervision ou tout au moins en déterminer les limites.
- ✓ Utilisateurs du SI :
  Ce questionnaire a pour but de dresser le profil des utilisateurs du SI. Ces informations vont permettre de se faire une idée de l'importance de l'indisponibilité du SI sur les utilisateurs. Ces utilisateurs vont être décrits comme suit :
  - Type — Les utilisateurs peuvent être internes (employés) ou externes (clients) à l'entreprise.
  - Localisation — S'il s'agit d'utilisateurs internes, le fait de spécifier leur localisation[39] va aider à cerner le comportement des utilisateurs en cas d'indisponibilité du SI. Ainsi, un

[39] Siège central, régional, agence, succursale à l'étranger, ...

utilisateur situé au guichet d'une agence, en contact avec la clientèle, va être plus difficilement enclin à accepter une indisponibilité du SI. Même si tous les utilisateurs du SI se trouvent dans un même département, l'indisponibilité de ce dernier peut provoquer la paralysie du département.

- Nombre — Ce nombre va donner une idée de grandeur de l'impact d'une indisponibilité du SI sur les utilisateurs.
- Rôle — Quel(s) rôle(s) jouent-ils dans l'organisation ? S'agit-il de "simples" encodeurs ou des utilisateurs dont l'activité est étroitement liée au métier de l'entreprise ? Cette information va permettre de pondérer l'importance d'un utilisateur pour l'entreprise. Par exemple, l'utilisateur du SI gérant les ordres de bourse sera certainement considéré comme plus stratégique que l'encodeur de virements bancaires.

✓ Volumes associés au SI

Toujours dans le but de se faire une idée de l'importance du SI, cette partie va permettre d'évaluer l'ordre de grandeur de l'activité et de la charge du SI. Trois éléments nous semblent importants :

- Le nombre moyen, maximum et minimum de transactions journalières va permettre d'estimer l'importance du flux de données transitant par le SI.
- Les périodes de pointes[40] vont aider à déterminer les périodes durant lesquelles, par exemple, les alertes de surcharge ne seront à prendre en compte.
- Le nombre d'éléments traités par le SI. En d'autres mots, quel est le nombre de clients, de comptes bancaires, de contrats, ... traités par le SI ? Cette donnée permet de quantifier l'importance du SI pour l'entreprise.

Contraintes du SI

Il s'agit d'examiner les éventuelles contraintes de temps, de sécurité ou contractuelles liées au SI. Cette information va permettre d'évaluer le type de support nécessaire ainsi que l'élaboration d'une supervision de composants autres que techniques, tels que la vérification de la disponibilité de données ou la fin d'exécution d'un programme dans un délai déterminé. Enfin, l'utilisation de programmes de cryptage, pour des raisons techniques, peut être un problème pour les outils de supervision. Les questions posées concernent :

✓ Heures de service

Quelles sont les heures de fonctionnement normales du SI ? Existe-t-il des périodes durant lesquelles ces heures sont différentes ? Cette information va permettre de déterminer le support adéquat. Un SI devant être disponible 24 heures sur 24 exigera un support de la *Master Console* pour les périodes situées en dehors des heures de bureau.

---

[40] Périodes de la journée, de la semaine, ... pendant lesquelles, par exemple, le nombre de transactions est nettement plus élevé que durant le reste du temps.

- ✓ Niveaux de qualité de service[41] interne
  Existe-t-il des accords concernant de tels niveaux avec d'autres départements de l'entreprise ? Ceci est important à savoir afin, notamment, de fixer les seuils d'alerte en accord avec ces niveaux.
- ✓ Niveaux de qualité de service externe
  Ceux-ci sont plus importants encore que les accords internes. En effet, dans la plupart des cas, ceux-ci sont fixés dans des accords contractuels et risquent, en cas de leur non-respect, de faire l'objet de sanctions financières. Par exemple, si la banque s'est engagée à effectuer des paiements dans un délai fixé, tout retard pourrait entraîner le paiement d'indemnités.
- ✓ Contraintes légales
  Le SI est-il soumis à des contraintes imposées par la loi (confidentialité, période de rétention des données, ...) ? Par exemple, les données comptables journalières doivent être disponibles avant la fin de la journée comptable. La supervision devra peut être inclure le contrôle de certains composants afin de s'assurer que le SI ne viole pas ces contraintes.
- ✓ Contraintes de timing
  Certaines opérations effectuées par le Si doivent-elles être terminées dans un délai déterminé ? Par exemple, certains transferts de données du SI doivent être effectués dans un temps imparti. Ces informations ont bien sûr un impact évident sur la supervision.
- ✓ Contraintes de sécurité
  Y a-t-il des contraintes liées à l'identification des utilisateurs, à la confidentialité, au contrôle des accès, à la non-répudiation[42] ? Dans ce cas, il faudra sans doute inclure dans la supervision des alertes concernant cette problématique.

Impact en cas d'indisponibilité

Il s'agit ici de mesurer l'impact d'une éventuelle indisponibilité du SI. Plus cet impact sera jugé important, plus bien sûr le projet de supervision risque d'être jugé prioritaire et le niveau de supervision élevé. L'impact est mesuré sur quatre critères :

- ✓ Utilisateurs
  L'indisponibilité peut avoir comme effet de rendre indisponibles pour les utilisateurs des données, des processus ou plus simplement des fonctionnalités. Que se passe-t-il donc pour l'utilisateur dans chacun de ces cas ? Est-il dans l'incapacité de travailler ou a-t-il d'autres moyens d'assurer sa mission ?
- ✓ Coûts
  Il s'agit de l'impact quantitatif et qualitatif de l'indisponibilité. Cela peut être une baisse de productivité, du chiffre d'affaires ou une perte de qualité des lignes de produits bancaires. Il est donc important de spécifier les lignes impactées par une indisponibilité du SI.
- ✓ Image de marque
  L'impact peut être vis-à-vis de la clientèle, de partenaires ou même de la concurrence.

---

41 Cela peut être très varié : temps de réponse, disponibilité minimum du SI, délai de mise à disposition de données, ...

42 La répudiation dans l'envoi de données informatiques est le fait, pour l'émetteur, de nier l'envoi (répudiation d'envoi) de ces données ou, pour le receveur, leur réception (répudiation de réception). Les mécanismes de non-répudiation sont donc des mécanismes qui empêchent toute contestation d'envoi ou de réception de données.

- ✓ Performances financières
  Il s'agit bien sûr du critère le plus important car le plus significatif pour les dirigeants de l'entreprise.

Contrôle du SI

Dans certains cas, les SI sont déjà dotés d'outils de supervision simples (commandes, programmes, ...) permettant un contrôle de base de leur fonctionnement. Il est utile de connaître l'existence de tels outils afin de pouvoir éventuellement les réutiliser. Dans ce cadre, deux questions seront posées :

- ✓ Contrôle technique
  Existe-t-il des outils, programmes ou commandes permettant de contrôler les composants techniques du SI ? En exemple, on peut citer des outils de gestion de base de données, de gestion de réseau, ...
- ✓ Contrôle métier
  Existe-t-il des outils, programmes ou commandes permettant d'assurer un contrôle de bout en bout du SI ? Existe-t-il des indicateurs associés au SI permettant de surveiller le comportement de ses fonctionnalités ?

Il est également important de demander au client quelles sont ses attentes en terme de supervision. En d'autres mots, quelles sont ses exigences pour la future supervision et quels indicateurs internes au SI il aimerait retrouver dans la supervision.

4.2.5.5 Participants

Aucun aspect technique n'est encore abordé dans cette activité. Outre la présence du membre de la cellule *Availability* pour assurer la présentation, celle du chef de projet de l'application à superviser suffit. Toutefois, afin de permettre une implication du correspondant technique dès le début du projet, sa présence est souhaitable dès cette activité.

Quant à la présentation qui est assez brève, elle peut éventuellement être omise ou remplacée par une explication donnée tout au long de la seconde partie de l'activité. Enfin, les deux parties de l'activité doivent, de préférence, se dérouler dans la foulée l'une de l'autre, au cours d'une même réunion, afin que les explications fournies restent profitables.

4.2.5.6 Résultats

A la fin de cette activité, le document instancié de la description logique du Si est produit.

4.2.6 Activité I.4 : Description technique du système d'information

4.2.6.1 Objectifs

Cette activité est destinée à obtenir de la part du client une description complète de l'architecture logique et technique du SI. Comme pour l'activité précédente, elle est divisée en deux parties :

- ✓ Une présentation d'explication du contenu du formulaire utilisé.
- ✓ Une réunion pour compléter le formulaire.

Comme pour l'activité précédente, plusieurs réunions et séances de travail seront sans doute nécessaires pour compléter le document dans sa totalité.

4.2.6.2 Contenu de la présentation

La présentation se déroulera comme suit :

Positionnement de l'activité	Le positionnement de l'activité consiste à restituer l'activité en cours dans tout le cycle du projet.
Résumé de l'activité	Le résumé de l'activité va présenter ce qui va être réalisé durant l'activité, les résultats produits ainsi que les personnes requises pour la deuxième partie de l'activité.
Explication du formulaire	L'explication du formulaire consiste à expliquer dans le détail le contenu du formulaire qui sera utilisé lors de la deuxième partie de l'activité. Il faut non seulement énumérer les informations dont la cellule *Availability* a besoin mais également expliquer pourquoi ces informations sont nécessaires.
Activité suivante	Enfin, pour conclure, l'activité suivante va donner un aperçu de la prochaine activité afin de donner au client un aperçu des informations qui lui seront demandées, des tâches à effectuer ou des personnes à convoquer pour la prochaine activité.

4.2.6.3 Documents utilisés

Pour la partie réunion de l'activité, les documents suivants sont nécessaires :

- ✓ Le guide utilisateur.
- ✓ Le formulaire utilisé durant l'activité réunion pour la collecte des informations relatives à la description technique du système d'information.

4.2.6.4 Formulaire de description technique du SI

Le formulaire utilisé pour la partie réunion de cette activité est destiné à obtenir une description technique logique et physique du SI. Il va permettre de dresser la topologie, représentation graphique des composants et de leurs connexions, du SI ainsi qu'une liste des composants et des logiciels utilisés. Le formulaire s'articule comme suit :

Description logique du SI

Pour obtenir une description logique la plus complète possible pour le projet de supervision, trois types d'informations sont nécessaires :

- ✓ Topologie logique du SI

  Une topologie logique est une topologie dans laquelle seront spécifiés les rôles fonctionnels (serveur de données, serveur d'applications, client, ...) de chaque composant. Cette topologie du SI est nécessaire afin de se faire une idée

générale de ses composants, de leur rôle fonctionnel ainsi que des relations qui existent entre eux. On spécifiera également si ce composant est présent en plusieurs exemplaires dans le SI.

- ✓ Relations logiques entre les composants
  On décrira les relations qu'il existe entre les différents composants. Celles-ci se décrivent comme suit :
  - Type de relation : Synchrone (lien transactionnel, appel de procédures stockées, *SQL*...), asynchrone (*MQSeries*, ...) ou simple transfert de fichier.
  - Direction : Quelle est la direction des échanges de données sur cette connexion ? Les échanges sont-ils unidirectionnels ou bidirectionnels ? Cette information est nécessaire afin de déterminer si la connexion doit être supervisée à partir d'un seul (l'émetteur) ou des deux composants de la connexion.

  Un graphique pourra également aider à se faire une idée plus précise des relations entre les composants.

- ✓ Relations logiques avec des SI externes
  Le type d'information demandé est identique à celui demandé pour les relations internes. On pourrait se demander pourquoi s'intéresser aux relations externes qui, à première vue, ne font pas partie intégrante du SI et du projet de supervision. Ce point est pourtant primordial et un exemple va le démontrer.

  Soit un SI de gestion de compte à vue via Internet pour la clientèle d'une banque. Pour obtenir le solde d'un compte, ce SI fait appel à une base de données centrale appartenant à un autre SI, le SI des comptes à vue par exemple. Dans ce SI sont supervisés tous les composants physiques et applicatifs internes du SI. L'indisponibilité de la base de données va bien sûr avoir un impact direct sur le bon fonctionnement du SI. La relation entre celui-ci et la base de données externe n'étant pas supervisée, aucune alarme n'est envoyée en cas d'indisponibilité des données et aucun problème ne sera donc détecté. Le SI de gestion de compte sera indiqué, malgré l'indisponibilité de base de données, comme étant pleinement opérationnel.

Description physique du SI

Cette partie est destinée à collecter un maximum d'informations sur les configurations techniques et logicielles des composants du SI. Il ne s'agit pas de définir les spécifications de la supervision mais bien de dresser un inventaire du matériel et des logiciels utilisés qui seraient susceptibles de faire l'objet d'une supervision. Pour chaque composant, les informations à fournir sont les suivantes :

- ✓ Nom du composant
  Identifiant unique de la machine. L'identifiant réseaux, par exemple. On ne spécifiera rien ici si la description du composant concerne un groupe de machines, par exemple, la description des plates-formes "client".

- ✓ Composant logique
  Rôle fonctionnel de ce composant (serveur de données, d'applications, client, ...). Cette information, reprise de la description logique du SI fournie lors de l'activité précédente, peut être utile afin de faire le lien avec cette description.

- ✓ Nombre d'instances
  Est le nombre d'instances d'un même composant au sein du SI (par exemple, le nombre de plates-formes "client"). Cette information est importante car si ce nombre est assez important, il conviendra d'être attentif à la fréquence d'envoi

des alarmes générées par ces composants. Trop d'alarmes risquent de surcharger le réseau ou d'engorger le serveur d'alarmes.

- ✓ Localisation
  Localisation du composant dans l'entreprise : siège central, siège régional, agence ou autre. Il est nécessaire de connaître cette localisation car les vitesses des lignes et les capacités des réseaux varient suivant cette localisation. Cela va du réseau local large bande pour le siège central au réseau à lignes téléphoniques à basse vitesse pour les agences. Cette information est donc nécessaire afin de limiter le nombre d'alarmes provenant des composants situés dans des réseaux lents.
- ✓ Zone de sécurité
  Le composant se trouve-t-il dans une zone sécurisée[43] ou un réseau interne ? Dans une zone sécurisée, de par les contraintes de sécurité, les possibilités de supervision sont très limitées. Il faudra donc informer le client de l'existence de telles limitations.
- ✓ Hardware
  Type de matériel du composant.
- ✓ Système d'exploitation
  Type et version du système d'exploitation installé sur les composants.
- ✓ Middleware
  Liste des logiciels installés sur le composant ainsi que les versions utilisées.
- ✓ Matériel spécifique
  Description du matériel particulier permettant de lui assurer une meilleure disponibilité tels que des disques redondants ou un système de protection contre les coupures de courant. La présence de tels systèmes va permettre notamment d'éliminer de la supervision la surveillance des ressources dont la disponibilité est assurée justement par de tels systèmes.
- ✓ Système en grappe
  Le composant est-il un système en grappe[44] ? Ce genre de système est plus compliqué à superviser car les applications peuvent indifféremment tourner ou, en cas de problème, basculer sur n'importe quelle instance de la grappe. La supervision devra donc être capable de faire la distinction entre une absence ou un arrêt normal d'une application d'un basculement ou d'un arrêt dû à un dysfonctionnement de l'application.

Ces informations sont à fournir pour les serveurs et éventuellement pour les stations de travail critiques du SI.

A noter que ces informations doivent être fournies aussi bien pour l'environnement de production que pour l'environnement de test. En fait, obtenir dès à présent ces informations pour l'environnement de test, va permettre, d'une part, d'éviter de contacter le client une seconde fois pour les collecter et, d'autre part, de sensibiliser le client sur le fait qu'un environnement complet de test, identique à l'environnement de production, est indispensable pour le projet de supervision.

Ressources systèmes

L'objectif de cette section est d'identifier, par composant logique du SI, toutes les ressources critiques qu'il serait opportun de superviser. Il est à noter que ne doivent faire partie de cet inventaire que les ressources absentes du catalogue et pour

[43] Voir Le catalogue page 66.
[44] Plus couramment appelé *Cluster*.

lesquelles aucune supervision n'existe. Les profils de supervision présents dans ce catalogue étant imposés et installés d'office sur les composants, ils ne peuvent faire l'objet de modification que dans le cadre d'un projet de mise à jour du catalogue. Les ressources peuvent être de plusieurs types :

- ✓ Programme
  Programmes systèmes, programmes de base d'un middleware utilisé ou programmes développés au sein du département informatique.
- ✓ Système de fichiers
  Tous les systèmes de fichiers du composant dont les propriétés[45] sont à surveiller.
- ✓ Fichiers
  Les fichiers critiques dont la présence, la taille et/ou le contenu sont à surveiller.
- ✓ Journaux électroniques
  Les journaux électroniques du système, des middleware ou des applications dans lesquels peuvent être trouvées des informations sur l'état de la ressource.

Ressources des relations

Il s'agit de dresser l'inventaire de toutes les ressources utilisées dans les relations tant internes qu'externes du SI. Ces ressources peuvent être du même type (programmes, fichiers,...) que les ressources système.

Transactions du SI

Il s'agit de décrire les éventuelles transactions "utilisateur" importantes[46], par exemple, de mise à jour ou de consultation, du SI. Pour chaque transaction, on indiquera :

- ✓ Nom
  Il peut s'agir d'un code ou du nom d'une transaction ou d'une procédure stockée.
- ✓ Contraintes
  Durée maximale d'exécution de la transaction, planification de son exécution, temps normal d'exécution, ...
- ✓ Volumes
  Volumes de données moyens traités par la transaction.
- ✓ Simulation
  Description d'un moyen technique de simulation de la transaction afin d'en tester ses performances ou sa disponibilité.

Toutes ces informations sont collectées dans l'éventualité où le niveau de supervision 5[47] serait choisi par le client.

4.2.6.5 Participants

Les formulaires portent principalement sur des questions d'architecture et de logiciels utilisés dans le SI. C'est pourquoi, en plus de la participation du chef de projet, celle du correspondant technique et de l'architecte du SI sont nécessaires.

---

[45] Taille, sécurité, attributs, ...
[46] Voir Niveaux de supervision page 56.
[47] Voir Niveaux de supervision page 56.

4.2.6.6 Résultats

A la fin de cette activité, deux documents instanciés sont produits :

- ✓ Le document reprenant la description technique du SI.
- ✓ L'extrait du catalogue reprenant tous les profils de supervision qui seront utilisés pour la supervision des composants ou d'une partie des composants du SI.

4.2.7 Activité I.5 : Spécifications fonctionnelles de la supervision

4.2.7.1 Objectifs

L'activité I.5 est une activité de type séance de travail ayant comme objectif de constituer les spécifications fonctionnelles de la supervision en fonction des données collectées dans la description technique du SI. Pour ces spécifications, il faudra :

- ✓ Eliminer les ressources dont la supervision est reprise dans l'extrait du catalogue.
- ✓ Définir les spécifications des autres composants, parties de composants ou ressources du SI.
- ✓ Eventuellement compléter les spécifications des composants techniques du catalogue avec des demandes de supervisions spécifiques.

Ces spécifications devront bien sûr tenir compte de l'architecture de supervision et de la découpe des profils de supervision[48].

Bien entendu, toutes les spécifications ne pourront être réalisées en une séance. Cette activité est donc itérative.

4.2.7.2 Documents utilisés

Quatre documents seront utilisés comme base de travail pour cette activité :

- ✓ Le guide utilisateur.
- ✓ L'extrait du catalogue des supervisions, produit lors de l'activité précédente, sera utilisé, d'une part, afin de sélectionner tous les profils de supervision standards à installer sur chacune des couches des différents composants pour lesquels une telle supervision existe et, d'autre part, afin d'enlever ces composants ou ces parties de composant de la phase de spécification.
- ✓ Le document de description technique du SI sera utilisé comme base d'inventaire des composants à superviser, donc des profils de supervision à constituer.
- ✓ Le formulaire de cette activité, destiné à la collecte des spécifications fonctionnelles de la supervision.

4.2.7.3 Le formulaire de spécification fonctionnelle

Ce formulaire est utilisé dans le but de collecter les spécifications fonctionnelles de la supervision. A noter que les informations nécessaires pour les spécifications varient suivant le niveau de supervision choisi par le client. Par exemple, pour le niveau de supervision 2, il sera nécessaire de compléter la partie du formulaire relative aux tâches associées aux ressources supervisées, ce qui n'est, par exemple, pas le cas pour le niveau 1. Enfin, s'il est nécessaire de définir une supervision générique et spécifique pour les composants présents en nombre dans le Si, des documents instanciés séparés seront produits pour les spécifications de la supervision générique et de chaque supervision spécifique.

---

[48] Voir Architecture de supervision page 49 et Profils de supervision page 52.

Le document sera subdivisé en trois parties :

- ✓ La spécification des ressources à superviser.
- ✓ L'établissement des règles de corrélation des alarmes.
- ✓ La spécification des tâches et des tâches personnalisées.

4.2.7.3.1 Spécifiation de la supervision des ressources

Les différentes ressources susceptibles d'être supervisées sont les suivantes :

- ✓ Les applications, programmes, services et processus.
- ✓ Les fichiers.
- ✓ Les systèmes de fichiers.
- ✓ Le contenu de journaux applicatifs.
- ✓ Les composants techniques tels que les disques, les cartes réseaux, ...
- ✓ Les transactions.

Applications

Pour chaque application, service ou processus à superviser, un minimum d'informations est nécessaire. Ces informations sont à fournir sous forme de tableaux à raison d'un tableau par ressource.

Supervision d'applications					
Machine		Application			
Zone de sécurité		Couche			
Planification					
Alarmes	Condition	Sévérité	Action(s)	ID	ENT
Journal					
Redémarrage					
Démarrage					
Arrêt					

Où : Machine — Identifiants de la machine ou du groupe de machines sur lesquelles le service, l'application ou le processus se trouve. Ce nom peut également être un type ou une famille d'un type de composants s'il s'agit de la spécification d'une supervision générique ou spécifique.

Application — Nom du programme, processus ou service à superviser.

Zone de sécurité — Localisation[49] du composant sur lequel se trouve l'application. Cette information est importante afin de déterminer si les moyens traditionnels de supervision peuvent être utilisés. Une zone sécurisée et un réseau externe, pour des raisons évidentes de sécurité, ne permettent pas l'utilisation de tels outils.

Couche — Couche de supervision[50] à laquelle appartient l'application. Ceci est nécessaire afin de ventiler, par couche, toutes les supervisions du

[49] Zone sécurisée, réseau interne, réseau externe, ...
[50] Voir Couches de supervision page 49.

SI. Ainsi, par exemple, si le SI à superviser est un middleware et qu'un programme du système d'exploitation doit être supervisé, un profil de supervision pour la couche "Système d'exploitation" sera créé pour ce programme ou sa supervision sera ajoutée à un profil existant.

Planification

Définit la fréquence à laquelle doit s'effectuer le contrôle de la disponibilité de l'application ainsi que les périodes durant lesquelles elle est requise. Par exemple, il peut être demandé de contrôler la disponibilité d'une ressource toutes les cinq minutes et ce, uniquement pendant les heures de bureau.

Alarmes

Décrit les différentes alarmes à générer en fonction de l'état ou du changement d'état de fonctionnement de l'application. Pour chaque alarme, il est nécessaire de spécifier :

- Condition : Condition de génération de l'alarme. Parmi les plus courantes, on peut citer : "est indisponible", "devient indisponible", "est disponible", "devient disponible", ... Suivant le niveau de supervision choisi, d'autres conditions plus spécifiques peuvent être utilisées : "utilise *x*% de la ressource *y*", "a *y* fichiers de type *x* ouvert", "a *y* connexions bloquées", ... Les possibilités sont bien sûr *quasi* illimitées.
- ID : Identifiant unique de l'alarme générée. Celui-ci sera être notamment utilisé dans les réseaux ERN.
- Sévérité : Degrés de sévérité assignée à l'alarme. Cette sévérité indique la gravité de la panne signifiée par l'alarme. Pour le degré de sévérité, il convient de respecter les standards décrits précédemment[51].
- Action(s) : Sont les actions à accomplir lors de la réception de l'alarme. Cela peut être, par exemple, l'exécution d'une tâche automatique telle que le redémarrage de l'application ou la génération d'un ticket d'incident.
- ENT : Spécifie si l'alarme doit ou non être envoyée à la *Master Console*. Cette option est directement liée au type de support désiré car seule la *Master Console* assure un support 24 heures sur 24 et 7 jours sur 7.

Journal

Nom et chemin d'accès complet du journal applicatif. Ce journal sera utilisé pour trouver des informations supplémentaires sur l'état de l'application ou comme base de supervision, au cas où l'utilisation des outils de supervision standards serait prohibée, comme dans une zone sécurisée par exemple.

Redémarrage

Si le redémarrage automatique figure dans la liste des actions, il est nécessaire de spécifier la commande à exécuter. Cette commande combine généralement arrêt/démarrage de l'application.

Démarrage

Commande de démarrage de l'application.

Arrêt

Commande d'arrêt de l'application. Ces deux dernières commandes peuvent être utilisées pour la définition des tâches du profil de supervision.

---

[51] Standardisation des sévérités page 58.

Fichiers

Il s'agit ici de spécifier la supervision à mettre en place, non pas sur le contenu du fichier mais sur l'état, le changement d'état ou de propriétés du fichier. Pour une telle spécification, les informations suivantes sont nécessaires :

Supervision de fichiers					
Machine		Fichier			
Zone de sécurité		Couche			
Planification					
Alarmes	Condition	Sévérité	Action(s)	ID	ENT

Où : Fichier Nom et chemin d'accès complet du fichier.

Alarmes Idem applications. Dans ce cas, les conditions les plus courantes sont : "Présent/Absent", "Permission d'accès a changé", "Taille change", "Taille maximum", "Taille minimum", "Taille augmente de *x* %"...

Machine, zone de sécurité, Couche, Planification Idem applications.

Attention : Afin de réduire au maximum le nombre d'alarmes générées dans les supervisions de capacité, de charge ou de taille de ressources, il est recommandé de suivre les deux principes suivants :

1. Seuils de valeurs : il convient de fixer comme suit les seuils des valeurs auxquelles les alarmes seront générées :
   - ✓ Un seuil pour une alarme d'avertissement.
   - ✓ Un seuil pour une alarme critique ou fatale. Ce seuil doit être nettement supérieur au seuil d'avertissement.
   - ✓ Un seuil pour un retour à la normale (fin d'incident). Ce seuil doit être nettement inférieur au seuil d'avertissement.

   En fixant de tels seuils, on disposera de toutes les alarmes nécessaires pour indiquer un léger dysfonctionnement du système (avertissement), une aggravation du problème menant éventuellement à une indisponibilité (critique/fatal) et enfin, d'une alarme indiquant la fin du problème (normal). Cette dernière clôturera automatiquement toutes les autres alarmes.

   Comme nous allons le voir dans le point 2, définir, par exemple, des seuils à 80 % pour une alarme d'avertissement et à 79 % pour un retour à la normale ou à 80 % pour une alarme d'avertissement et à 81 % pour une alarme critique ou fatale, est vivement déconseillé.

2. Génération des alarmes : afin de limiter le nombre d'alarmes générées, il convient de respecter les règles suivantes :
   - ✓ Une alarme est générée une seule fois lors du dépassement d'un seuil. Ainsi, si un seuil est dépassé durant une période couvrant plusieurs test de la ressource, seule la première détection du dépassement donnera lieu à la génération d'une alarme.
   - ✓ Une alarme d'avertissement est envoyée lorsque le seuil est atteint ou dépassé ET que la ressource se retrouve dans un état de fonctionnement normal.

- ✓ Une alarme critique/fatale est envoyée lorsque le seuil est atteint ou dépassé ET que l'on se trouve dans un état avertissement.
- ✓ Pour l'état normal, une alarme n'est envoyée que lorsque la capacité, la charge ou la taille de la ressource supervisée est retombée sous le seuil de retour à la normale.

Dans l'exemple ci-après, représentant la charge d'un processeur, les seuils des alarmes normales, avertissement et critique/fatal ont été respectivement fixés à 60, 80 et 90 %. Dans cet exemple, trois alarmes seront générées. Elles sont représentées par un carré jaune pour l'alarme d'avertissement, rouge pour l'alarme critique/fatale et vert pour la normale.

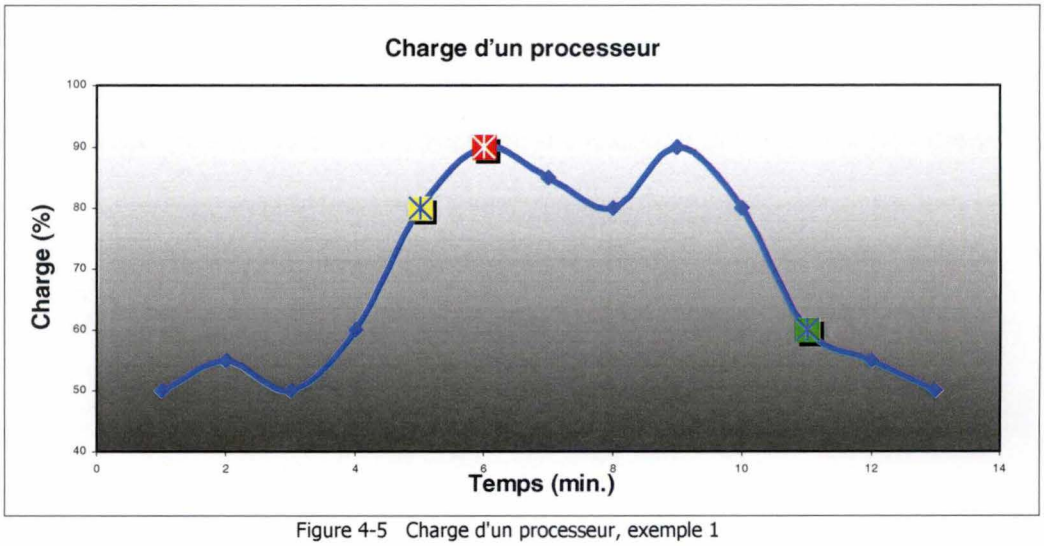


Figure 4-5 Charge d'un processeur, exemple 1

On pourrait se poser la question de savoir pourquoi fixer le seuil de retour à la normale nettement plus bas que le seuil d'avertissement. En effet, si 80 % est le seuil d'avertissement, pourquoi 79 % ne conviendrait-il pas pour un retour à l'état normal ? L'exemple suivant va en expliquer la raison.

Soit la supervision d'une charge d'un processeur. Si les seuils d'alarmes sont fixés comme dans l'exemple précédent, les alarmes seront générées comme suit :

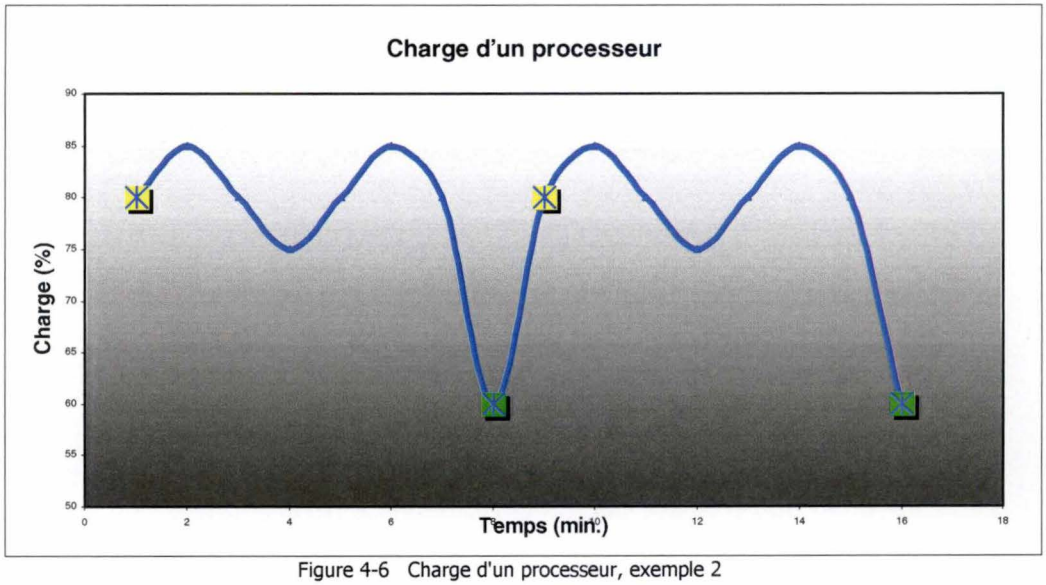


Figure 4-6 Charge d'un processeur, exemple 2

Soit maintenant le même graphique de charge mais dont le seuil de retour à la normale à été fixé à la même valeur du seuil d'avertissement. Dans ce cas, les alarmes seront générées comme suit :

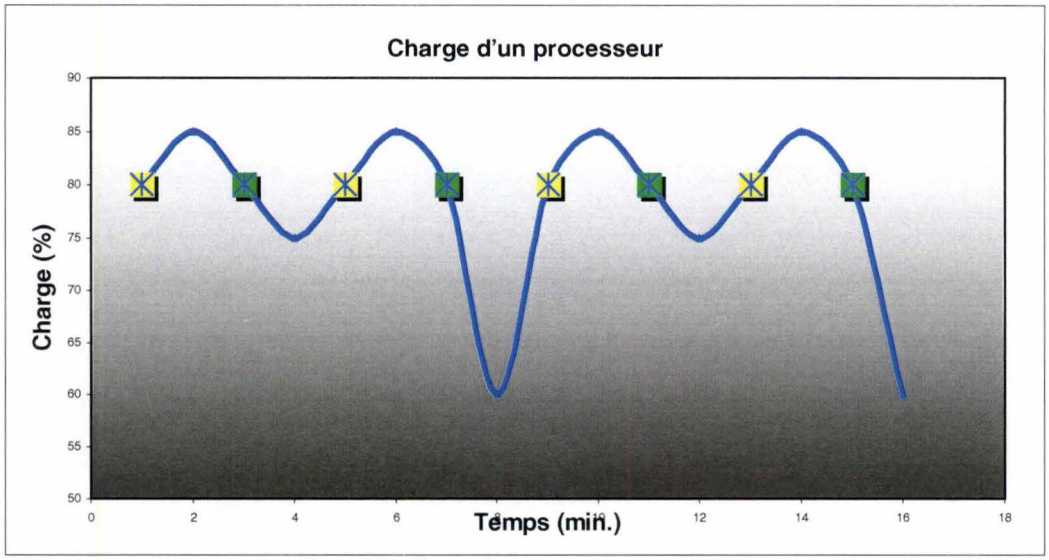


Figure 4-7 Charge d'un processeur, exemple 3

On remarque directement que le nombre d'alarmes envoyé est nettement plus important. Le respect de cette règle très simple permet de limiter le nombre d'alarmes générées et ainsi d'éviter une éventuelle surcharge du réseau ou du serveur d'alarmes.

Systèmes de fichiers

Un système de fichiers est un moyen pour une plate-forme de disposer de disques de données distants tout en donnant l'impression à ce système que ces disques lui sont propres. Par exemple, un *Filesystem* pour *Unix* ou un disque logique réseau pour *NT*.

Il s'agit ici de spécifier la supervision d'un système de fichiers. Pour une telle spécification, les informations suivantes sont nécessaires :

Supervision de systèmes de fichiers					
Machine		Système			
Point de montage					
Zone de sécurité		Couche			
Planification					
Alarmes	Condition	Sévérité	Action(s)	ID	ENT

Où : Nom de machine Idem applications.

Système Nom du système de fichiers.

Point de montage Nom de la machine où réside le système de fichiers et nom réel de ce système.

Alarmes Idem fichiers. A noter cependant que la responsabilité quant à l'accessibilité (montage) d'un système de fichiers, qu'il serve pour le système d'exploitation, un middleware ou une application,

dépend toujours de l'équipe technique responsable de la plate-forme concernée. Une telle supervision devra, si elle est nécessaire, être ajoutée au profil de supervision de la couche "Système d'exploitation" pour le type de cette plate-forme.

Zone de sécurité, couche et planification Idem applications.

Contenu de journaux applicatifs

Lors de la spécification de la supervision des applications, il a fallu fournir le nom ainsi que l'endroit où l'on pouvait trouver le journal lié à l'application. Un tel journal est habituellement utilisé par une application pour garder une trace de son activité et notamment des éventuels dysfonctionnements qu'elle a rencontrés. Pour la supervision, il est possible de surveiller le contenu de tels fichiers afin de générer des alarmes lorsque certains messages sont présents. Pour une telle supervision, les informations suivantes sont nécessaires :

Supervision des journaux applicatifs					
Machine		Application			
Zone de sécurité		Couche			
Planification					
Message		Sévérité	Action(s)	ID	ENT

Où : Nom de machine Idem applications.

Son nom Nom et chemin d'accès du fichier.

Alarmes Idem applications sauf pour la condition qui n'a pas de sens ici. A la place, il faudra spécifier le message du fichier pour lequel il faut générer une alarme.

Zone de sécurité, couche et planification Idem applications.

Certaines applications parmi les plus récentes utilisent des messages à format structuré dans leurs journaux applicatifs. Dans ce cas, il est intéressant de décrire ce format sous forme de masques. Un masque est une description générique d'un type de message du journal. Il peut donc y avoir plusieurs masques pour un même fichier. Chaque masque devra être décrit comme suit :

- ✓ Une découpe d'un message d'exemple en champs.
- ✓ Une description du type de valeurs que l'on peut trouver dans chaque champ du message.
- ✓ Pour chaque champ, spécifier si la valeur de celui-ci est variable ou fixe.

Exemple :

Message	18 Oct	15:24:21	UT124	Login : ROOT	Type3	DE	Nova	Critique
Masque	Date	Heure	N° Utilisateur	Texte	N° terminal	Texte	Nom machine	Sévérité
Type	Variable	Variable	Variable	Fixe	Variable	Fixe	Variable	Variable

Composants techniques

Pour les supervisions techniques, il peut être nécessaire de superviser des composants tels que le processeur, les disques physiques, des cartes réseaux, ... Ce genre de supervision sera exclusivement implémenté dans des profils de supervision appartenant à la couche de

supervision hardware[52]. Pour la spécification de telles supervisions, les informations suivantes sont nécessaires :

Supervision de composant technique					
Machine		Composant			
Zone de sécurité		Couche			
Planification					
Alarmes	Condition	Sévérité	Action(s)	ID	ENT

Où : Machine Idem applications.

Composant Type du composant à surveiller.

Alarmes idem applications. Parmi les conditions les plus utilisées, on trouve : "est/devient indisponible", "est/devient disponible" ainsi que des seuils d'occupation ou de taux d'erreurs.

Zone de sécurité, couche, planification Idem applications.

Transactions

Comme nous l'avons vu dans les niveaux de supervision[53], le niveau 5 inclut une supervision de bout en bout de l'application. Un tel niveau de supervision tente de refléter la vue qu'a un utilisateur réel sur le SI en termes de disponibilité de ses fonctionnalités et de performances. Généralement, une telle supervision peut être réalisée par :

- ✓ L'utilisation d'outils spécialisés.
  Ces outils mettent à disposition des programmeurs des interfaces qui peuvent être exécutées par les transactions. Ces interfaces permettent de calculer les performances à toutes les étapes d'une transaction ainsi que son cheminement dans le SI. Ce genre d'outil utilise des données réelles mais présente le grand désavantage que son utilisation doit être implémentée, et prévue dès le départ, dans les transactions réelles du SI.
- ✓ La programmation de simulateurs.
  Ces simulateurs exécutent à intervalles réguliers des transactions réelles du SI, en retirent des conclusions sur la disponibilité des systèmes utilisés et sortent des statistiques sur les performances de cette transaction. Le principal désavantage de tels outils réside dans la nécessité de prévoir des données fictives, susceptibles d'être modifiées par la simulation. Comme ces données sont toujours les mêmes, elles ne représentent pas un échantillon représentatif des données du SI. En effet, étant souvent consultées ou modifiées, elles peuvent éventuellement bénéficier de mécanismes propres aux systèmes permettant des consultations ou modifications rapides de données fréquemment consultées. Ces mécanismes risquent ainsi de fausser les résultats obtenus par les simulateurs qui ne refléteront plus la réalité.
- ✓ L'utilisation d'informations directement disponibles dans le SI.
  Certains SI gardent, généralement dans des journaux applicatifs, des messages relatifs aux performances et à la disponibilité de ses transactions. Cette supervision peut être réalisée par une consultation classique des journaux.

[52] Voir Responsabilités de supervision page 50.
[53] Voir Niveaux de supervision page 56.

Indépendamment de la manière dont sera implémentée cette supervision, il convient de décrire toutes les transactions qui seront susceptibles d'être supervisées. Pour ce faire, les informations suivantes sont nécessaires :

Supervision de transactions					
Machine client					
Zone de sécurité					
Application		Transaction			
Utilisateur					
Simulateur		Paramètres			
Planification					
Alarmes	Condition	Sévérité	Action(s)	ID	ENT

Où :

Machine client — Identifiant de la machine à utiliser comme client dans la simulation de la transaction.

Zone de sécurité — Idem applications. Il s'agit de la zone du serveur et non du client utilisé pour la simulation.

Application — Nom de l'application ou des applications qui utilisent la transaction.

Transaction — Nom ou code de la transaction.

Utilisateur — Identifiant d'un utilisateur sous l'autorité duquel la transaction doit être exécutée.

Simulateur — S'il existe, le nom du programme de simulation de la transaction.

Planification — Intervalles et calendrier d'exécution de la simulation de la transaction.

Paramètres — Eventuels paramètres à fournir pour l'exécution du simulateur.

Alarmes — Idem applications. Comme exemple de condition, on peut citer : "temps moyen de la transaction supérieur à ", "a échoué/a réussi", "nombre d'échecs supérieurs à"...

4.2.7.3.2 Spécification des règles de corrélation

Les règles de corrélation des alarmes sont les règles qui régissent le flux des actions qui seront exécutées, lors de la réception d'une alarme, sur une ou plusieurs autres alarmes. Dans ces spécifications, il s'agit de décrire toutes les relations qui existent entre ces alarmes. Ces règles vont découler sur un processus qui va s'exécuter exclusivement au niveau du serveur d'alarmes *T/EC*. Comme relation entre alarmes, on peut notamment citer :

- ✓ Clôture d'une ou plusieurs alarmes par une ou plusieurs autres alarmes.
- ✓ Dégradation (augmentation de sévérité) d'une alarme.
- ✓ Incrémentation d'un compteur lors de la réception d'alarmes identiques.
- ✓ Génération d'une nouvelle alarme.

✓ Abandon d'une alarme jugée non significative[54].

Pour spécifier ces règles de corrélation, la méthodologie EMD/EMCD[55] offre un formalisme intéressant pour les raisons suivantes :

1. Il est universel en ce sens qu'il n'est pas lié à un outil de supervision spécifique et peut être utilisé pour toutes les alarmes quelle qu'en soit la source.
2. Il est simple. Il ne demande pas d'expertise particulière pour être appliqué.
3. Les réseaux de relations des alarmes (ERN[56]) sont une base documentaire très efficace. Ils permettent en un coup d'œil d'avoir une vue d'ensemble des relations qui existent entre les alarmes.

Certaines aspects de cette méthodologie nous semblent cependant superflus ou doivent être adaptés pour répondre parfaitement aux besoins de la cellule *Availability*. Reprenons ces cinq ateliers de la méthodologie EMD/EMCD dans le détail en y apportant les modifications nécessaires.

Atelier 1 Sélection des sources d'alarmes.

La ou les sources des alarmes que l'on doit traiter sont déjà connues puisqu'il s'agit d'alarmes générées par le SI. Cet atelier n'est plus nécessaire et est donc supprimé.

Atelier 2 Inventaire des référentiels d'alarmes.

L'inventaire des alarmes est déjà réalisé dans sa presque totalité mais est éparpillé dans les différents tableaux de spécification. Il peut être utile de regrouper toutes ces alarmes au sein d'un seul tableau d'autant plus que cette liste servira de base aux activités ultérieures. Le tableau utilisé pour cette activité se présentera dorénavant comme suit :

ID	Nom	Description	Source

Où : ID Est l'identifiant de l'alarme issu des spécifications.

Nom Est le libellé de l'alarme ou, par exemple, le message issu d'un journal applicatif.

Description Est une brève description, une explication de l'alarme.

Source Est l'origine de l'alarme, c'est-à-dire la ressource qui a généré l'alarme.

Atelier 3 Décision de filtrage des alarmes.

Cet atelier n'a plus de raison d'être. En effet, les alarmes générées par la supervision sont toutes significatives puisqu'elles sont tous générées en réponse à des situations déterminées par le client. L'activité de filtrage est devenue superflue.

Atelier 4 Analyse pour la corrélation des alarmes.

Pour notre propos, le tableau associé à l'atelier se présente ainsi :

ID	Nom	Candidat à corrélation	Détection alarmes identiques	ERN	Sévérité

[54] Par exemple, parce qu'une alarme de sévérité supérieure est déjà présente.
[55] Voir Event Management Design (EMD) page 33.
[56] *Event Relationship Network.*

Où : ID et Nom Représentent les même données que dans le tableau précédent.

Candidat à corrélation La valeur est "OUI" si l'alarme intervient dans une corrélation. La valeur est "NON" si l'alarme est autonome. A noter que, typiquement, une alarme autonome sera à clôturer manuellement au niveau du serveur d'alarmes.

Détection alarmes identiques Si sa valeur est "OUI", on ne garde et ne montre qu'un seul exemplaire de cette alarme au niveau du serveur d'alarmes. Si des alarmes identiques arrivent par la suite, un compteur sera incrémenté afin d'indiquer le nombre total reçu. Dans le cas contraire ("NON"), toute alarme identique reçue sera gardée et montrée.

ERN Est le nom du réseau de relation d'alarme dans lequel est décrite la corrélation à laquelle appartient l'alarme.

Sévérité Est la sévérité assignée à l'alarme[57].

Atelier 5 Elaboration des réseaux de relations d'alarmes

Ce type de réseaux est à garder car il représente très clairement la corrélation. Il est facile à lire et on a, en un coup d'œil, une vue générale d'une corrélation. Il faut cependant apporter une petite modification de présentation. Dans le réseau original, le code couleur est défini comme suit :

- ✓ **Vert** pour les alarmes de clôture (*Clearing Event* dans la méthodologie)
- ✓ **Rouge** pour les alarmes primaires et secondaires (*Primary* et *Secondary*).
- ✓ **Jaune** pour les alarmes secondaires/primaires (*Secondary*/ *Primary*).

Ce choix n'est pas optimum pour deux raisons :

1. Associer des couleurs aux types d'alarme n'est pas nécessaire puisque ce type est déjà indiqué dans la bulle d'alarme (P, S, C, S/P).
2. Il n'est pas possible de voir les incohérences de corrélation.

Pour justifier le point 2, un exemple s'impose.

Soit trois alarmes EV_1, EV_2, et EV_3 avec leurs sévérités respectives Avertissement, Critique et Fatal. La corrélation suivante a été dessinée :

[57] Voir Standardisation des sévérités page 58.

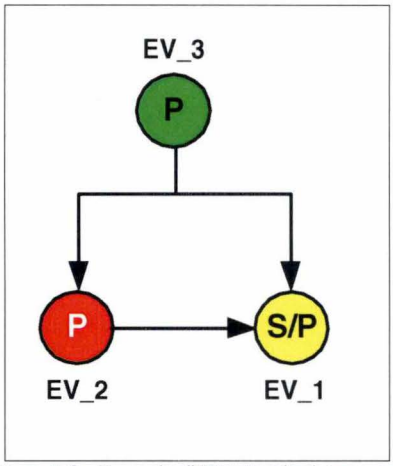


Figure 4-8 Exemple d'ERN standard avec problème

Deux incohérences se sont glissées dans ce réseau :

1. L'alarme EV_3 est de sévérité "Fatal" mais est considérée dans la corrélation comme un incident de clôture ce qui n'est pas logique car une alarme de clôture doit être de sévérité "Normal".
2. L'alarme EV_1 est de sévérité inférieure à l'alarme EV_2. Or, elle est représentée comme une alarme conséquente (aggravation du problème) de celle-ci. La couleur représentant le type et non la sévérité de l'alarme, une telle erreur risque de passer inaperçue.

Ces deux erreurs probablement dues soit à une mauvaise assignation des sévérités soit à une erreur dans la conception de la corrélation, ne sont pas directement visibles à la lecture du réseau. Ce problème peut être résolu en assignant à chaque bulle d'alarme non plus la couleur du type d'alarme mais bien la couleur indiquant le degré de sévérité[58].

Le réseau se présentera dès lors comme suit :

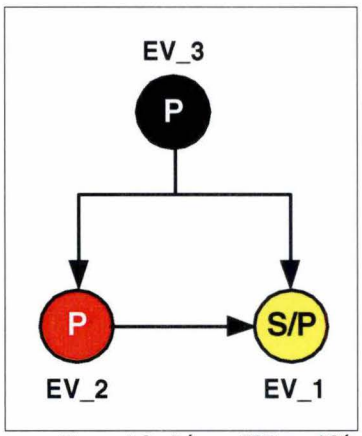


Figure 4-9 Réseau ERN modifié

Les couleurs représentant maintenant les degrés de sévérité, les deux erreurs sont à présent détectées facilement et peuvent être éventuellement corrigées.

A noter enfin que les identifiants des alarmes utilisés dans les réseaux sont tirés des tableaux des spécifications fonctionnels[59].

[58] Voir Spécifiation de la supervision des ressources page 85.
[59] Voir Spécifiation de la supervision des ressources page 85.

4.2.7.3.3 Spécification des tâches et des tâches personnalisées

Si un niveau de supervision autre que le niveau 1 est choisi, il convient de spécifier les tâches associées aux différents niveaux intermédiaires. Comme nous l'avons vu[60], il existe deux types de tâches, les tâches et les tâches personnalisées. Pour chacun de ces types, un tableau de spécification spécifique sera utilisé.

Pour la spécification des tâches, les informations suivantes sont nécessaires :

Nom	Programme	Autorisation		Description
		Utilisateur	Groupe	

Où : Nom — Nom donné à la tâche.

Programme — Nom du programme qui sera réellement exécuté.

Autorisation — Identifiant d'utilisateur (ainsi que du groupe fonctionnel auquel il appartient) à utiliser pour l'exécution de la tâche. Cette information est nécessaire dans le cas où des droits de sécurité spécifiques seraient nécessaires pour l'exécution de cette tâche.

Description — Est une description des fonctionnalités de la tâche.

Pour les tâches personnalisées, les informations suivantes sont nécessaires :

Nom	Tâche	Cibles	Paramètres	Commentaire

Où : Nom — Nom donné à la tâche personnalisée.

Tâche — Nom de la tâche universelle utilisée.

Cibles — Cible(s) sélectionnée(s).

Paramètres — Valeurs associées À TOUS les paramètres de la tâche.

Commentaire — Justification de la création de la tâche personnalisée.

4.2.7.4 Participants

Toutes les informations nécessaires pour cette activité sont d'ordre technique. La présence du correspondant technique suffit donc.

4.2.7.5 Résultats

A la fin de l'activité, le document instancié reprenant les spécifications fonctionnelles de la supervision est complet.

---

[60] Voir Les outils de supervision page 19.

4.2.8 Conclusion

A la fin de cette phase, tous les documents produits sont regroupés dans un dossier fonctionnel qui sera soumis à l'acceptation du client. Ce dossier comprend :

- ✓ La description du SI.
- ✓ La description technique du SI.
- ✓ Le document de spécifications fonctionnelles.
- ✓ La sélection des profils extraits du catalogue.

Ce dossier va servir de base à la phase II du projet de supervision : l'étude de faisabilité.

4.3 Phase II : Etude de faisabilité

4.3.1 Objectifs

L'étude de faisabilité est la phase durant laquelle les spécifications fonctionnelles seront traduites en spécifications techniques. Ces spécifications techniques vont couvrir :

- ✓ La validation de l'étendue de la supervision.
- ✓ La validation des spécifications fonctionnelles.
- ✓ La découpe éventuelle du projet en plusieurs phases de développement.
- ✓ L'évaluation de la charge de travail nécessaire au développement de la solution de supervision.
- ✓ La traduction des spécifications fonctionnelles en spécifications techniques.
- ✓ La planification du développement, des tests et du déploiement de la supervision.

4.3.2 Plan de la phase

La phase II se décompose comme suit :

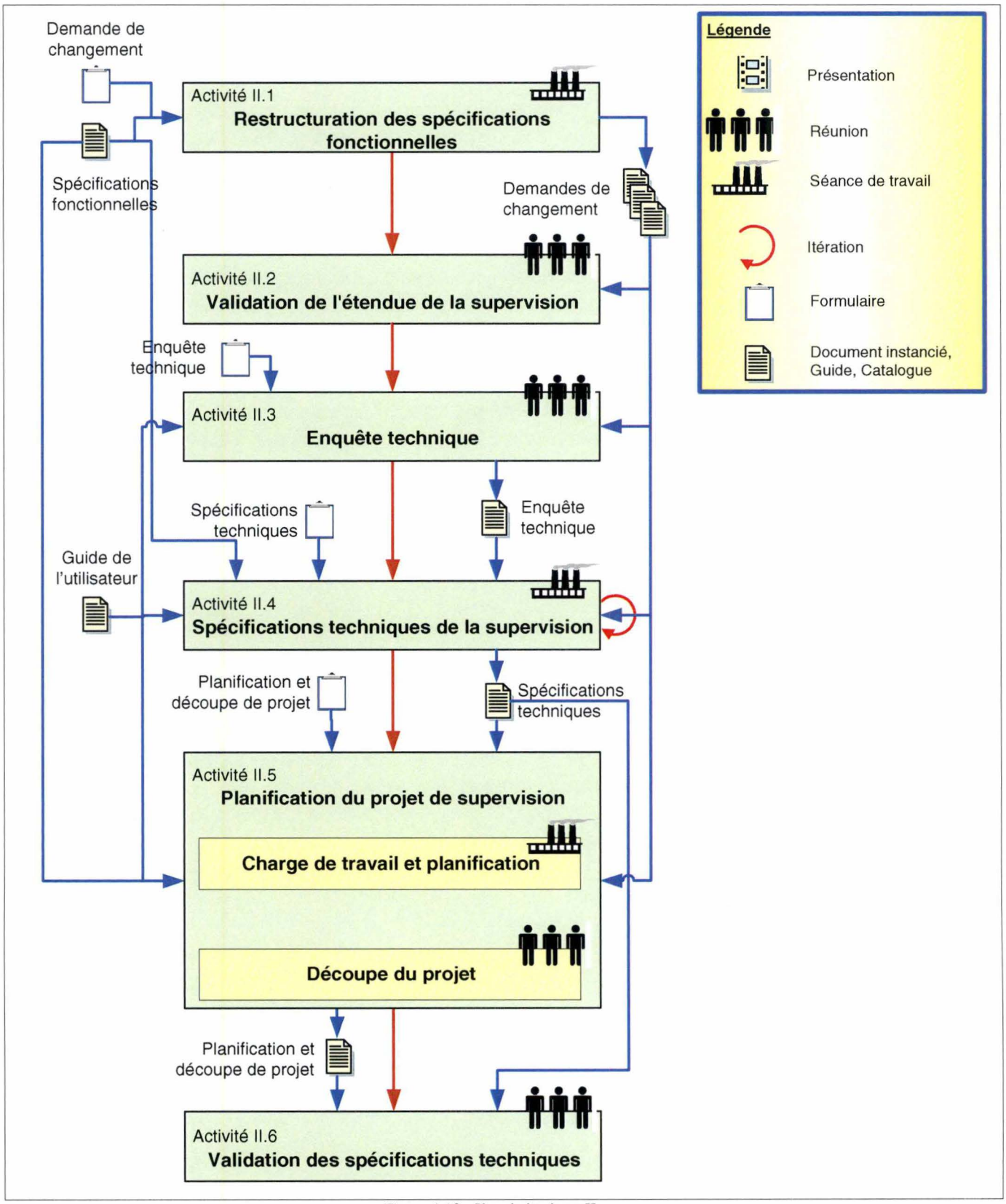


Figure 4-10 Plan de la phase II

4.3.3 Activité II.1 : Restructuration des spécifications fonctionnelles

4.3.3.1 Objectif

La restructuration des spécifications fonctionnelles est une activité de type séance de travail qui consiste à reprendre le dossier des spécifications fonctionnelles et d'en :

- ✓ Eliminer toutes les supervisions de ressources dont la responsabilité n'incombe pas à la couche de supervision à laquelle appartient le SI. Par exemple, si le SI supervisé est un *middleware*, il conviendra d'éliminer toutes les supervisions de ressources dont la responsabilité de supervision n'incombe pas à la couche *Middleware*[61].
- ✓ Ventiler ces supervisions au sein de profils de supervision existants ou dans de nouveaux profils de supervision et soumettre pour acceptation ces demandes de changement aux départements techniques[62].

4.3.3.2 Elimination de la supervision

Dans l'architecture de supervision que nous avons mise en place, des frontières de responsabilité pour la supervision ont été fixées pour chaque couche. Ces frontières définissent la liste des ressources que chaque couche est autorisée à superviser. Dans ce cadre, il convient donc d'épurer le dossier fonctionnel des spécifications de toutes les supervisions de ressources dont la responsabilité incombe à une autre couche. Après élimination, ne resteront, dans ce dossier, que les spécifications de supervision ayant trait à l'application elle-même.

Exemple :
Soit un projet de supervision d'un SI dont les ressources suivantes sont à surveiller. Aucune de ces supervisions n'est présente dans le catalogue :

- ✓ Trois programmes applicatifs.
- ✓ Deux fichiers applicatifs.
- ✓ Deux programmes systèmes.
- ✓ Un système de fichiers dont l'accessibilité et l'espace libre sont à surveiller.
- ✓ Des fichiers d'une base de données.

Sachant que la couche à laquelle appartient le SI est la couche 'Application', il convient d'éliminer de ces spécifications toutes les supervisions de ressources dont la responsabilité n'incombe pas à cette couche. La liste épurée des spécifications de supervision précédente se présentera dès lors comme suit :

- ✓ Trois programmes applicatifs.
- ✓ Deux fichiers applicatifs.
- ✓ L'espace libre du système de fichiers.

En effet, la responsabilité de la supervision de programmes systèmes ainsi que l'accessibilité d'un système de fichiers incombent à la couche "Système d'exploitation"[63]. La supervision des fichiers de la base de données, quant à elle, incombe à la couche "*Middleware*"[64].

---

[61] Voir Responsabilités de supervision page 50.
[62] Voir Les acteurs page 63.
[63] Voir Responsabilités de supervision page 50.
[64] Voir Responsabilités de supervision page 50.

4.3.3.3 Ventilation des supervisions

Les spécifications ainsi extraites du dossier fonctionnel vont faire l'objet de négociations avec les différents départements techniques afin de :

- ✓ Les inclure dans un profil de supervision existant au cas où le type de composant serait déjà sujet à une supervision et donc présent dans le catalogue.
- ✓ Faire l'objet d'un nouveau profil de supervision si aucun profil de supervision n'existe pour le type de composant concerné.

Si les spécifications sont à inclure dans un profil de supervision existant, le choix doit être fait, soit de les inclure dans un profil générique, soit de créer un profil de supervision spécifique[65]. Cette décision appartient exclusivement au département technique responsable de l'administration du composant. En effet, ce département est le mieux à même de juger si une ressource peut être supervisée uniformément ou si elle est spécifique à une plate-forme, une version du composant ou à un domaine fonctionnel particulier. Pour reprendre le cas des fichiers de la base de données de l'exemple précédent[66], il est évident que les administrateurs du système de base de données concerné sont les personnes les mieux placées pour juger de l'opportunité d'une telle supervision et, si tel est le cas, de savoir si ces fichiers sont propres au système de base de données du SI ou s'ils sont systématiquement présents dans tous les systèmes de même type. En d'autres mots, la ressource est-elle spécifique au SI ou générique ? La réponse à cette question va déterminer le type de profil de supervision dans lequel sera placée la supervision de la ressource. Une supervision de ressource générique sera versée dans le profil de supervision générique du composant concerné tandis que la supervision d'une ressource propre au SI sera placée dans le profil de supervision spécifique correspondant ou, éventuellement, va mener à la création d'un tel profil.

Enfin, si aucun profil de supervision, tant générique que spécifique, n'existe pour ce composant, en d'autres mots, si ce dernier ne fait pas encore partie du catalogue de supervision, il conviendra dès lors d'entamer, en collaboration avec les départements techniques, la construction de tels profils. Ces profils devront, dans la mesure du possible, reprendre les spécifications du client.

Toutes ces demandes de modifications ou de créations de profils de supervision seront soumises aux départements techniques concernés via des documents instanciés de demande de changement.

4.3.3.4 Acceptation des demandes de changement

L'acceptation des demandes de changement consiste pour les départements techniques à évaluer ces demandes et à prendre une décision quant au choix d'implémenter ou non la demande et, dans le cas d'une décision positive, de déterminer la meilleure manière de le faire. Ces départements doivent également déterminer s'il s'agit d'une supervision générique à un type de composant ou spécifique au seul composant concerné.

4.3.3.5 Formulaire de demande de changement

Le formulaire de demande de changement est utilisé afin d'inclure ou de modifier la supervision d'une ressource dans un profil générique ou spécifique ou comme demande de création d'un profil de supervision spécifique. Ce formulaire se présente comme suit :

Description du changement

---

[65] Voir Supervision générique et spécifique page 53.
[66] Voir Elimination de la supervision page 100.

Il s'agit de décrire l'objet de la demande de changement. Cette description est subdivisée en deux parties :

- ✓ Un résumé destiné à replacer la demande dans son contexte. Ce résumé reprend :
  - La date de la demande.
  - L'auteur de la demande, en l'occurrence la cellule *Availability*.
  - L'identification de la demande. Les demandes suivent la convention de nom suivante : DCxxxx avec ***DC*** pour 'Demande Changement' et ***xxxx***, un numéro attribué séquentiellement par projet aux demandes de changement.
- ✓ Une description détaillée de la demande de changement. Cette description comprend quatre sections et regroupe les informations suivantes :
  - Le champ d'application de la demande. Ce champ d'action décrit le composant, le type de composant ou le groupe de composants pour lequel cette demande est introduite.
  - Une description de la supervision. Celle-ci peut reprendre le même formalisme[67] que dans le document de spécification.
  - Les contraintes liées à la demande. Ces contraintes peuvent être d'ordre divers : délai à respecter, priorité (criticité) des ressources supervisées.
  - Les projets susceptibles d'être influencés par la demande. En ajoutant des supervisions susceptibles de se retrouver dans des profils génériques, des nouvelles alarmes risquent d'apparaître pour des projets plus anciens. Pour permettre une étude d'impact de ce changement, il convient de dresser la liste des projets concernés.

Implémentation du changement

La deuxième partie du document décrit la suite qui sera donnée à la demande de changement. Cette partie sera complétée par le département technique auquel la demande sera adressée. Elle reprend les informations suivantes :

- ✓ Décision — Décision prise pour la demande. La demande peut être :
  - Acceptée. La supervision de la ressource sera donc soit ajoutée dans un profil de supervision existant, soit assurée par un profil de supervision à créer pour assurer cette dernière.
  - Refusée. Le département technique juge la supervision de la ressource inacceptable parce qu'elle est entre autres, inutile ou dangereuse[68]. Par exemple, de par son implémentation, *UNIX* utilise toujours 100 % de son processeur. Superviser la charge du processeur n'a donc aucun sens.
  - Reportée. Le département technique juge la demande acceptable mais préfère reporter sa décision afin d'étudier le meilleur moyen d'implémenter la supervision ou de définir une solution plus globale dans le cas, par exemple, d'une demande de supervision pour un composant n'étant encore soumis à aucune supervision. C'est notamment le cas pour les composants absents du catalogue.

---

[67] Sous forme de tableaux. Voir Activité I.5 : Spécifications fonctionnelles de la supervision page 84.

[68] Parce qu'elle risquerait, par exemple, de mettre en péril la stabilité du composant ou de générer trop d'alarmes.

- ✓ Justification — Dans le cas d'un refus ou d'un report, listes des raisons qui ont amené à la décision.
- ✓ Implémentation — Dans le cas d'une acceptation, la partie consacrée à l'implémentation va indiquer le type d'implémentation de la modification demandée et où elle le sera. Le type d'implémentaton décrira la meilleure méthode à utiliser pour la supervision de la ressource : outils standards, programmes spécifiques, outils spécialisés, ... L'implémentation indiquera également le nom ainsi que le type[69] de profil dans lequel sera ajouté la supervision ou le profil qu'il sera nécessaire de créer pour assurer la supervision.
- ✓ Responsable — Identifie le correspondant technique qui a pris la décision quant à la suite à donner à la demande de changement.
- ✓ Date — Date de la prise de décision de la demande de changement.

4.3.3.6 Participants

La partie de l'activité consacrée à l'élimination des supervisions est uniquement assurée par la cellule *Availability*. C'est elle qui sera chargée d'épurer le dossier des spécifications fonctionnelles et d'introduire toutes les demandes de changement pour les supervisions ainsi extraites du dossier fonctionnel. La décision quant à la suite qui sera donnée à ces demandes incombe, quant à elle, aux différents départements techniques concernés.

4.3.3.7 Résultats

A l'issue de cette activité seront produits :

- ✓ Une série de demandes de changement pour les supervisions faisant partie du catalogue. Certaines sont acceptées, d'autres seront mises en attente ou refusées. Ces demandes feront l'objet d'un projet séparé au fur et à mesure de leur acceptation.
- ✓ Le dossier fonctionnel épuré et restructuré qui sera utilisé lors de la prochaine activité.

4.3.4 Activité II.2 : Validation de l'étendue de la supervision

4.3.4.1 Objectif

Cette activité de type réunion a pour objectif de valider l'étendue de la supervision qui sera mise en place. A cette fin, la cellule *Availability* va :

- ✓ Attirer une nouvelle fois l'attention du client sur les limites de la supervision, c'est-à-dire mettre l'accent, d'une part, sur d'éventuelles ressources ou connexions qui ne sont pas soumises, volontairement ou involontairement, à la supervision et, d'autre part, sur les limitations des moyens de supervision pour certaines ressources dues à la localisation[70] ou aux particularités techniques[71] de la ressource.

---

[69] Générique ou spécifique.
[70] Zone sécurisée, réseaux lents, ...
[71] Système de disques redondants, système en grappe, ...

- ✓ Soumettre à la validation du client le dossier fonctionnel épuré produit lors de l'activité précédente.

Cette activité n'a donc comme autre objectif que d'ôter toute ambiguïté sur les limites de la future supervision. Elle doit mener à l'acceptation par le client des spécifications fonctionnelles de la supervision.

4.3.4.2 Participants

Cette réunion impliquera un représentant de la cellule *Availability*, le correspondant technique ainsi que le chef de projet qui est, pour la cellule *Availability*, le seul habilité à accepter la solution.

4.3.4.3 Résultats

Le dossier fonctionnel de la supervision sera validé et accepté par la cellule *Availability* et par le client.

4.3.5 Activité II.3 : Enquête technique

4.3.5.1 Objectif

L'objectif de cette enquête est de dresser un inventaire des outils propres au SI susceptibles d'aider à la supervision des ressources spécifiées dans le dossier fonctionnel. En effet, en dehors des ressources traditionnelles tels que les programmes ou les fichiers, il peut être demandé de superviser des ressources spécifiques au SI comme des files d'attentes ou des zones tampons. Il n'est pas possible d'assurer ce genre de supervision via les outils standards de supervision. Ces derniers peuvent cependant faire appel à ces outils spécialisés afin d'obtenir l'information nécessaire sur l'état de telles ressources.

4.3.5.2 L'activité

On dénombre trois grandes catégories d'outils susceptibles d'être utilisés par les outils de supervision :

- ✓ Outils de gestion
  Il s'agit de programmes spécialisés de gestion d'infrastructure tels que les outils de gestion de réseau ou de base de données. Il est intéressant de savoir si ces outils peuvent être configurés pour envoyer directement des alarmes souhaitées vers le serveur d'alarmes.
- ✓ Langage de commande[72]
  Il s'agit d'un ensemble de commandes permettant d'interroger le composant (programmes, logiciels ou middleware) sur l'état de ses ressources internes.
- ✓ Les interfaces "programme"[73]
  Ces interfaces permettent d'inclure dans des programmes des appels directs aux programmes, logiciels, ... afin d'obtenir des services ou des informations internes de fonctionnement de ces logiciels. Par exemple, l'interface programme de *Windows* permet de construire et d'afficher des fenêtres, des boutons, des graphiques, ...

---

[72] Aussi appelé CLI : *Common Language Interface.*
[73] Communément appelées API : *Application Program Interface.*

Afin de collecter toutes les informations nécessaires, un formulaire spécifique est utilisé.

4.3.5.3 Le formulaire d'enquête technique

Pour chaque outil, il convient d'obtenir les informations suivantes :

- ✓ Description de l'outil
  Il s'agit de décrire l'outil susceptible d'être utilisé en fournissant les informations suivantes :
  - Nom — Nom de l'outil.
  - Type — CLI, API ou outil de gestion.
  - Version — Version de l'outil utilisé.
  - Compatibilité — Type et version des plates-formes sur lesquelles l'outil peut être utilisé.
- ✓ Mode d'exécution
  Explique comment utiliser l'outil, la syntaxe des commandes, des appels systèmes, du serveur de commande. Cette partie décrit également les paramètres éventuels à utiliser.
- ✓ Prérequis
  Décrit les programmes additionnels, les autorités d'accès, la configuration technique, ... minimum nécessaire pour utiliser l'outil.

4.3.5.4 Participants

Les informations à fournir pour cette activité sont d'ordre purement technique. C'est pourquoi, outre la présence du représentant de la cellule *Availability*, celle du correspondant technique est requise.

4.3.5.5 Résultats

A la fin de cette activité, le document instancié tiré du formulaire de l'enquête technique est produit.

4.3.6 Activité II.4 : Spécifications techniques de la supervision

4.3.6.1 Objectifs

L'activité de spécification technique de la supervision est une activité de type séance de travail qui consiste à :

- ✓ Spécifier les tests à effectuer pour déterminer chaque condition de génération d'une alarme.
- ✓ Traduire les réseaux de relations d'alarmes[74] en valeurs de corrélation. Il s'agit ici en fait de décrire le contenu des champs des alarmes utilisés par le processus de corrélation.

La charge de travail de cette activité est assez importante. Il est évident qu'elle sera itérative.

---

[74] Voir Spécification des règles de corrélation page 92.

4.3.6.2 Description de l'activité

La définition des tests pour les conditions de génération des alarmes consiste à définir, pour chacune de ces conditions, la procédure à suivre pour la déterminer. Par exemple, si l'on désire envoyer des alarmes lorsqu'une application X est "disponible" et "indisponible", il faut déterminer ce que signifient ces deux états pour cette application X. Dans la plupart des cas, tester la simple présence d'un ou plusieurs services ou programmes suffira. Cependant, dans le cas de supervision de ressources bien spécifiques (disponibilité de sous-composants tels que les files d'attentes, les connexions, les zones tampons...), des tests plus complexes peuvent être nécessaires. Si les outils standards de supervision ne permettent techniquement pas d'effectuer ces procédures, il est nécessaire de spécifier, pour chaque type, les outils à utiliser ainsi que la manière de les utiliser. Pour le choix des outils et des méthodes de supervision à utiliser, on veillera à respecter les deux règles suivantes :

- ✓ Pour les raisons déjà évoquées[75], respecter l'ordre de préférence des types de supervision établi, à savoir :
  1. Supervision intégrée.
  2. Supervision pro-active.
  3. Supervision réactive.
- ✓ Afin de minimiser le développement de nouveaux moniteurs, on veillera également à respecter l'ordre d'utilisation des outils suivant :
  1. Outils standards de supervision. Ces outils sont prêts à l'emploi et ne demandent aucun développement supplémentaire. Ils sont cependant limités à la supervision de ressources "traditionnelles" telles que les fichiers, les services ou les programmes.
  2. Outils de supervision existants. Dans certains cas, les responsables des SI ont développé des outils de supervision sur mesure. Ils sont généralement très simples mais permettent facilement de surveiller l'état de fonctionnement interne du SI. Afin de ne pas réinventer la roue, il convient de voir comment ces programmes peuvent être réutilisés ou modifiés pour d'obtenir les informations nécessaires pour la supervision.
  3. Outils spécifiques. Si ceux-ci existent, il convient de les utiliser et éventuellement de les améliorer afin qu'ils envoient directement les alarmes nécessaires.
  4. Nouveaux outils. En dernier recours, il faudra développer de nouveaux programmes spécifiques. Ces programmes pourront ou non faire appel aux outils spécifiés lors de l'activité précédente.

Dans le processus de corrélation, la relation entre une alarme de type *Clearing* et les alarmes *Primary* ou *Primary / Secondary* du réseau de relations est une relation de clôture d'alarme. Techniquement, elle va se traduire par la spécification de deux champs dans les alarmes :

- ✓ Ferme : liste des alarmes *Primary* ou *Primary / Secondary* qui seront clôturées par cette alarme.
- ✓ Fermé_Par : liste des alarmes *Clearing* susceptibles de clôturer cette alarme.

L'arrivée d'une alarme "Clearing" va donc clôturer toutes les alarmes contenant dans le champ *Fermé_Par* une valeur identique présente dans son champ *Ferme.* De plus, pour éviter que des alarmes de clôture ne restent ouvertes, on spécifiera dans les champs Fermé_Par qu'elles sont clôturées par elles-mêmes.

La relation entre des alarmes *Primary* et *Primary / Secondary* est, quant à elle, une relation de cause à conséquence. Elle se traduit par la spécification des champs :

---

[75] Voir Types de supervision page 52.

- ✓ Est_Cause : liste des alarmes étant la conséquence de cette alarme.
- ✓ Est_Conséquence : liste des alarmes étant la cause de cette alarme.

Pratiquement une telle relation est traduite par le fait que la clôture ou l'acquittement[76] d'une alarme de cause implique la même action sur toutes ses alarmes conséquences.

On constate que pour ce processus de corrélation, il est nécessaire d'identifier toutes les alarmes de manière unique. Il va donc falloir imaginer un système d'identification des alarmes permettant cette identification non équivoque. L'identification doit permettre de distinguer une alarme :

1. A d'une alarme B.
2. A_1 du composant C_1 de la même alarme A_1 pour un composant C_2, quelle que soit la localisation de ces ressources[77].
3. A_1 de la ressource R_1 du composant C de la même alarme A_1 de la ressource R_2 du même composant C ou d'un composant différent.

Exemple :

Soit deux systèmes *MQSeries* **MQ_1** et **MQ_2** sur une plate-forme **PF**. Chacun de ces deux systèmes possèdent deux connexions **C_1** et **C_2**.

Ces deux systèmes peuvent envoyer les alarmes suivantes :

- ✓ Pour l'application **MQ_x**: ETAT_MESSAGERIE pour signaler son arrêt ou son démarrage avec un champ indiquant son état (disponible ou indisponible).
- ✓ Pour les connexions **C_x** : CONNEXION_ETABLIE pour signaler qu'un connexion **C_x** est établie, CONNEXION_PERDUE pour signaler que la connexion s'est terminée et REESSAI_CONNEXION lorsque l'application **MQ_x** essaie à intervalles réguliers de rétablir une connexion **C_x**.

Si l'on reprend les 3 points précédents, il faut pour cet exemple pouvoir distinguer une alarme :

1. CONNEXION_PERDUE d'une alarme ETAT_MESSAGERIE ou d'une alarme REESSAI_CONNEXION...
2. ETAT_MESSAGERIE pour le système MQ_1 de la même alarme pour le système MQ_2.
3. CONNEXION_PERDUE pour la connexion C_1 du système MQ_1 de la même alarme pour la connexion C_2 de ce même système ou du système MQ_2.

Pour répondre à ces 3 points, chaque alarme sera identifiée de la manière suivante :

{Libellé_Alarme}_{Nom_Plate-forme}[@{Nom_Composant}[@{Nom_Ressource}]] [_{Etat}]

Avec :

{Libellé_Alarme}	Texte de l'alarme.
{Nom_Plate-forme}	Identifiant de la plate-forme sur laquelle se trouve le composant.
{Nom_Composant}	Identifiant du composant.

[76] Voir Notions de supervision page 19.
[77] Sur une même ou des plates-formes différentes.

{Etat} Etat du composant (par exemple "disponible") dans le cas où le libellé de l'alarme serait unique pour tous les états du composant ou de la ressource.

{Nom_Ressource} Identifiant de la ressource.

Si l'on reprend l'exemple précédent, les identifiants des alarmes *ETAT_MESSAGERIE* et *CONNEXION_PERDUE* se présenteront comme suit :

ETAT_MESSAGERIE_PF@MQ$_1$_DISPONIBLE

Pour une alarme indiquant la disponibilité du composant MQ_1 sur la plate-forme PF.

CONNEXION_PERDUE_PF@MQ$_1$@C$_1$

Pour une alarme indiquant la perte de la connexion C1 du système MQ1 situé sur la plate-forme PF.

Si l'on considère qu'une alarme CONNEXION_ETABLIE clôture les deux alarmes CONNEXION_PERDUE et REESSAI_CONNEXION, les champs de corrélation pour ces alarmes de la connexion C_1 du composant MQ_1 se présenteront comme suit :

Pour une alarme CONNEXION_ETABLIE :

Ferme = CONNEXION_PERDUE_PF@MQ$_1$@C$_1$, ESSAI_CONNEXION_PF@MQ$_1$@C$_1$, CONNEXION_ETABLIE_PF@MQ$_1$@C$_1$

Fermé_Par = CONNEXION_ETABLIE_PF@MQ$_1$@C$_1$

Pour une alarme CONNEXION_PERDUE ou ESSAI_CONNEXION :

Fermé_Par = CONNEXION_ETABLIE_PF@MQ$_1$@C$_1$

Comme nous l'avons vu précédemment dans l'élaboration des corrélations[78], il peut être demandé de ne visualiser qu'une seule instance d'une alarme lorsque celle-ci est susceptible d'être reçue en plusieurs exemplaires et de compter le nombre de copies reçues. Par exemple, on peut recevoir, à intervalles de temps réguliers, une alarme ESSAI_CONNEXION indiquant que le système tente régulièrement de rétablir une connexion et n'afficher que la première apparition et compter le nombre de tentatives suivantes. A cette fin, le processus de corrélation utilise le champ *Clé_Identique* sur lequel il se base pour détecter les duplications d'alarme. La valeur de ce champ doit bien entendu suivre les mêmes règles d'unicité que les identifiants utilisés pour les champs de corrélation. La même convention de nom peut donc être utilisée ici aussi.

En fin d'activité, il conviendra de spécifier toutes les tâches personnalisées ou non, spécifiées dans le dossier fonctionnel.

4.3.6.3 Formulaire de spécification technique

Les spécifications techniques de la supervision sont à regrouper dans un document dont le formulaire se découpe comme suit :

Pour chaque ressource à superviser, on spécifiera :

- ✓ Plate-forme — nom de la ou des plates-formes sur laquelle ou lesquelles trouve la ressource à superviser.
- ✓ Nom — nom de la ressource.

[78] Voir Spécification des règles de corrélation page 92.

- ✓ Méthode — description de la méthode choisie pour l'implémentation de la supervision de la ressource. A savoir : outil spécialisé, interface programme, nouveau programme, ...
- ✓ Implémentation — description détaillée de l'implémentation de la supervision. Celle-ci peut être faite en pseudo-code pour les nouveaux programmes, une liste de commandes à exécuter, ...

Toutes les alarmes qui seront générées devront être décrites dans le tableau suivant :

Spécifications techniques des alarmes					
ID		Libellé			
Champs	Valeurs	Remplis	Support	SCIM	URL

Où ID — Identifiant de l'alarme. Cet identifiant est tiré des spécifications fonctionnelles de la supervision des ressources[79].

Libellé — Le libellé de l'alarme.

Champs — Liste des champs présents dans l'alarme. Dans cette liste, on retrouvera notamment les champs de corrélation *Ferme*, *Fermé_Par*, *Est_Cause*, *Est_Conséquence* et le champ de détection des doublons *Clé_Identique*.

Valeurs — Valeur contenue dans les différents champs.

Remplis — L'endroit où sera rempli le champ. *Source* si le champs est rempli directement à la source de l'alarme, *T/EC* s'il l'est lors de la réception de l'alarme sur le serveur d'alarmes. A noter que, afin de décharger un maximum ce serveur, il est conseiller de compléter, dans la mesure du possible, un maximum de champs à la source.

Support — Si la génération d'un ticket d'incident fait partie des actions prévues[80] pour l'alarme, ce champ contient l'adresse de destination du ticket, c'est-à-dire l'adresse du groupe responsable de la résolution du problème. Une destination de ticket est l'adresse d'une boîte aux lettres électronique assignée, par exemple, à un groupe fonctionnel, une cellule de département ou encore un groupe de support et à laquelle sera envoyée une notification d'arrivée de ticket.

SCIM — Utilisé par *TSD*, il décrit le problème grâce à quatre niveaux d'information, du plus général au plus précis. Ces niveaux sont les suivants :

- ✓ **S**ystème — décrit le système. Par exemple, un système d'exploitation, un type de matériel, ...
- ✓ **C**omposant — indique le type de système.
- ✓ **I**tem — indique la ressource qui a provoqué la panne de ce système. Cela peut être une application, un disque physique ou logique, un fichier...
- ✓ **M**odule — indique, par exemple, un code ou un message d'erreur ou encore l'identification d'un module en erreur.

[79] Voir notamment Le formulaire page 84
[80] Voir Le formulaire de spécification fonctionnelle page 84.

Par exemple, pour un problème sur une imprimante :

Système	Imprimante
Composant	HP Laser Jet IV
Item	Entraînement papier
Module	Erreur 13

URL — adresse d'une page HTML dans le site Intranet de documentation technique du département informatique. Sur cette page, la *Master Console* trouvera toutes les informations nécessaires à la résolution du problème. Cette information est obligatoire pour toute alarme destinée à la *Master Console*[81].

Pour chaque tâche et tâche personnalisée, il conviendra de spécifier les informations suivantes :

- ✓ Plate-forme — nom de la ou des plates-formes pour lesquelles la tâche est destinée.
- ✓ Nom de la tâche — nom que l'on va donner à la tâche.
- ✓ Paramètres — description des éventuels paramètres requis par la tâche.
- ✓ Prérequis — description de l'environnement requis pour une bonne exécution de la tâche : disponibilité de données, version de logiciels, espace mémoire, ...
- ✓ Implémentation — description de la tâche. Le pseudo-code et un formalisme intéressant pour ce genre de description.

4.3.6.4 Participants

Le correspondant technique peut à lui seul être apte à répondre à toutes les questions du formulaire. Cette activité sera une collaboration entre ce dernier et le représentant de la cellule *Availability*.

4.3.6.5 Résultats

A la fin de cette activité, le document instancié contenant les spécifications techniques est complet.

4.3.7 Activité II.5 : Planification du projet de supervision

4.3.7.1 Objectifs

L'activité de planification se décompose en deux parties, l'une de type séance de travail, l'autre de type réunion. Elle a comme objectifs de :

- ✓ Estimer la charge de travail et définir une planification de la supervision.
- ✓ Proposer au client une découpe du projet en plusieurs phases afin de mettre en production le plus rapidement possible des différentes parties de la supervision au fur et à mesure de leur disponibilité.

---

[81] Dont la colonne ENT des spécifications fonctionnelles de l'alarme est rempli. Voir Spécifiation de la supervision des ressources page 85.

4.3.7.2 Charge de travail et planification

Le calcul de la charge de travail consiste à estimer le nombre de jours/homme que vont nécessiter les développements, les tests et la mise en production des programmes de supervisions et des tâches éventuelles de la supervision. Bien que nous entrions ici dans une problématique de gestion de projet plus traditionnelle, moins intéressante pour notre propos, il est néanmoins utile d'attirer l'attention sur les points suivants :

- ✓ Les moniteurs sont des programmes qui s'installent sur les plates-formes où résident les ressources qu'ils supervisent. Pour certains types de plates-formes, l'installation des programmes quels qu'ils soient s'effectue via un outil de distribution de logiciels. Les demandes de distribution sont envoyées à un département spécialisé qui les traite en fonction de sa charge de travail ou de l'urgence de la demande. Afin de permettre une planification la plus correcte possible, il convient de fixer avec ce département un SLA[82] pour la prise en compte et l'exécution des demandes de distribution de moniteurs.
- ✓ Sur d'autres plates-formes, les mises à jour des applications s'effectuent par lot à date fixe dans l'année à raison de quatre mises à jour par an. Tout comme ces applications, les moniteurs à installer sur ces machines doivent être inclus dans ces lots. La mise en production de la supervision sur de telles plates-formes peut facilement être retardée de plusieurs semaines, voire même de plusieurs mois. La planification du projet de supervision doit tenir compte de ce paramètre.
- ✓ Comme nous l'avons vu, la restructuration des spécifications fonctionnelles peut générer une série de demandes de changement des supervisions présentes dans le catalogue ou même mener à un nouveau projet de supervision. Ici aussi, il convient de négocier avec les départements techniques des SLA qui fixent les délais pour les réponses aux demandes de changement et pour la mise à disposition des spécifications d'implémentation de ces supervisions. Pour les supervisions qui seront prises en compte, elles seront menées dans des projets parallèles menés en fonction des disponibilités de la cellule *Availability*. Il convient de les retirer de la planification puisque aucune échéance ne peut encore être fixée.

4.3.7.3 Découpe du projet

Dans la plupart des cas, il ne sera pas possible dans un laps de temps raisonnable de mettre en place la solution de supervision dans sa totalité. Plusieurs paramètres peuvent influencer la période de mise en œuvre de la supervision :

- ✓ Le niveau de supervision[83] choisi
  Plus ce niveau est élevé, plus le nombre de ressources à superviser et le nombre de tâches à écrire risque d'être important.
- ✓ Le manque de couverture du catalogue des supervisions
  Toutes les supervisions présentes dans le catalogue sont disponibles immédiatement et peuvent donc être installées sans délai. Elles ne nécessitent aucun effort de développement. Plus le nombre de composants du SI se retrouvant dans le catalogue est important, plus la charge de travail pour la mise en œuvre de la supervision sera réduite.

---

[82] *SLA* : *Service Level Agreement*. Accord, par exemple, sur une qualité de service ou un délai maximum d'exécution de tâche.

[83] Voir Niveaux de supervision page 56.

- ✓ La complexité technique
  Il est évident que plus la supervision est complexe, plus la charge de travail sera difficile et importante.
- ✓ La présence ou l'absence d'outils de supervision
  L'utilisation de tels outils peut fortement réduire les développements nécessaires.
- ✓ La dépendance vis à vis de la gestion des versions
  Comme nous l'avons vu, sur certaines plates-formes techniques, les mises à jour des applications s'effectuent par lot à date fixe dans l'année. Il convient de regrouper les développements des programmes devant être inclus dans ces lots.
- ✓ L'hétérogénéité des plates-formes techniques
  Plus il y a de composants techniques différents, plus les développements seront nombreux. Aussi, il devra même parfois être nécessaire d'écrire un même programme plusieurs fois suivant les plates-formes présentes.

Afin d'éviter que de tels problèmes ne bloquent la totalité du projet de supervision, il peut être opportun de morceler le projet de supervision en plusieurs étapes. Celles-ci seront définies de manière à mettre le plus rapidement possible à la disposition du client un maximum de supervision.

Pour cela, les découpes suivantes peuvent être proposées au client :

- ✓ Découpe par niveau de supervision.
  Si un niveau de supervision élevé (5 par exemple) est choisi, on peut proposer de diviser le projet de supervision à raison d'une étape par niveau de supervision. Si cette méthode de découpe offre l'avantage de diviser la charge de travail, elle ne tient pas compte des exigences liées aux installations de programmes sur les plates-formes. Cela peut avoir un impact direct sur la planification du projet.
- ✓ Découpe par type de plate-forme.
  L'idée est de regrouper les développements liés à la supervision par type de plate-forme. Cette solution permet de centraliser les efforts sur un type de plate-forme à la fois et permet d'avoir, pour ce type, une supervision complète en une seule fois. Elle présente cependant le danger de rendre les développements moins universels avec le risque de devoir adapter, voire même de recommencer, les développements pour d'autres plates-formes.
- ✓ Découpe par disponibilité de supervision.
  Il s'agit ici de découper le projet en fonction des délais de développement ainsi que des exigences des mises en production des plates-formes techniques. Il s'agit donc de regrouper les développements par paquets en fonction de la charge de travail qu'ils représentent et de leur dépendance vis-à-vis de la gestion des versions. Très logiquement, dans ce type de découpe, on installera d'abord toutes les supervisions tirées du catalogue. Celles-ci sont d'ores et déjà approuvées et sont directement disponibles. Elles peuvent être activées immédiatement. Viennent ensuite les différents paquets de programmes.

De par la complexité des ressources qui les composent ou leur caractère critique, chaque projet de supervision est différent. La méthode de découpe du projet à adopter sera à déterminer pour chaque cas. Bien qu'il n'y ait pas de découpe à privilégier par rapport à une autre, nous pensons cependant que la troisième méthode répond davantage aux besoins tant de la cellule *Availability* que du client à savoir, obtenir une supervision opérationnelle, même partielle, dans les délais les plus courts.

4.3.7.4 Formulaire de découpe du projet

Un formulaire unique peut être utilisé pour la découpe du projet et la planification des différentes étapes ainsi dégagées. Ce formulaire ne contient rien de bien spécifique à la problématique de la supervision. Un formulaire de planification de projet classique, dont la description n'apporterait rien dans l'étude qui nous occupe, peut être utilisé sans problème.

4.3.7.5 Participants

Si la planification du projet relève de la responsabilité de la cellule *Availability*, la découpe éventuelle du projet en plusieurs étapes appartient au client. Aussi, pour cette partie de l'activité, la présence du chef de projet du Si s'avère nécessaire.

4.3.7.6 Résultats

A l'issue de cette activité, les documents instanciés de planification et de découpe du projet sont disponibles.

4.3.8 Activité II.6 : Validation des spécifications techniques

4.3.8.1 Objectif

L'objectif de cette activité est d'obtenir de la part du client, la validation des spécifications techniques établies lors de la phase II. Il s'agit de reprendre les spécifications techniques de la supervision et de voir si celles-ci répondent toujours tant techniquement que fonctionnellement aux besoins du client.

4.3.8.2 Participants

Pour cette phase, la présence du chef de projet et du correspondant technique est nécessaire. Le premier est requis pour valider la solution de supervision dans son ensemble, le second pour valider les solutions techniques retenues pour la supervision de chaque ressource.

4.3.8.3 Résultats

A la fin de cette activité, les documents instanciés reprenant les spécifications techniques sont validés tant par le client que par la cellule *Availability*.

4.3.9 Conclusion

Tous les documents produits lors de cette phase seront regroupés pour former le dossier technique de la supervision. C'est sur ce dossier que se baseront tous les développements propres à la supervision.

4.4 Phase III : Développement de la supervision

4.4.1 Objectifs

La phase III concentre toutes les activités de développement liées à la supervision. Elle comprend :

- ✓ La définition des moniteurs.
- ✓ L'écriture des moniteurs.
- ✓ L'écriture des tâches.
- ✓ La mise en place de l'architecture propre aux produits de supervision.
- ✓ L'écriture des règles de corrélation.
- ✓ La rédaction des pages Intranet d'aide à la résolution des problèmes utilisés par la *Master Console*.

Si ces développements ne nécessitent pas vraiment d'étude particulière dans le cadre de la méthodologie, il est cependant intéressant de s'attarder sur deux formulaires utilisés durant cette phase, à savoir :

- ✓ Le formulaire pour la documentation technique des programmes.
- ✓ Le formulaire des pages Intranet.

Enfin, durant toute la phase de développement, des activités de synchronisation seront régulièrement organisées afin de faire de point sur l'état d'avancement des développements et de prendre des décisions quant aux problèmes éventuels.

4.4.2 Plan de la phase

Mis à part les réunions régulières de synchronisation, il n'y a pas d'activité particulière dans cette phase. Celle-ci est uniquement vouée aux développements de la supervision.

4.4.3 Activité de synchronisation

L'activité de synchronisation est une activité de type réunion. En fait, il ne s'agit pas d'une seule mais d'un cycle de réunions durant lesquelles seront abordés les problèmes et questions relatifs aux développements ou au design même de la supervision ainsi que les problèmes relatifs au contrôle du respect des délais. C'est également durant ces réunions que le client pourra introduire d'éventuelles demandes de changement relatives à la supervision. Pour de telles demandes, le même formulaire utilisé pour les révisions des supervisions du catalogue[84] peut être repris sans aucune modification.

Ces réunions réunissent un représentant de la cellule *Availability* et le correspondant technique du client. La fréquence de ces réunions est bien sûr à déterminer entre ces deux parties.

4.4.4 Formulaire de documentation technique des programmes

Ce formulaire est utilisé afin de documenter de la manière la plus complète possible les programmes écrits pour la supervision. Il sera utilisé indifféremment pour la documentation des moniteurs, des tâches ou des programmes d'installation de l'architecture de supervision propre au projet.

---

[84] Voir Formulaire de demande de changement page 101.

Ce formulaire se présente comme suit :

Introduction

L'introduction consiste à décrire le cadre d'utilisation du programme. En d'autres mots, il s'agit de décrire brièvement le projet de supervision, la ressource supervisée ou la tâche pour laquelle a été écrit le programme.

Objet

L'objet contient une brève description des fonctionnalités du programme.

Résumé

Le résumé va dresser une description technique du programme : ses différentes versions, ses prérequis, les plates-formes supportées, ... Il comprend sept sections :

- ✓ Versions
  Historique des différentes modifications apportées au programme lors de ses différentes versions. Pour ce faire, le tableau suivant sera utilisé :

Version	Date	Modifications

Où : Version — Numéro de la version.
Date — Date de la mise à jour.
Modifications — Description des modifications apportées au programme depuis dans cette version.

- ✓ Responsabilité
  Département, cellule, équipe ayant la responsabilité de maintenir et de modifier le programme. Cette responsabilité est décrite dans le tableau suivant :

Département	Cellule	Contact	Remarques

Où : Département — Nom et numéro du département responsable.
Cellule — Nom et numéro de la cellule ou de l'équipe de ce département.
Contact — Nom d'une ou plusieurs personnes servant de point de contact pour tout renseignement concernant le programme.
Remarques — Remarques éventuelles sur le mode de contact, la particularité de la responsabilité, ...

- ✓ Définition
  Type du programme, c'est-à-dire s'il s'agit d'un programme destiné à la supervision d'une ressource, d'une tâche, d'une tâche personnalisée, d'installation d'architecture, ...

- ✓ Langage
  Langage de programmation et version de ce langage utilisé pour ce programme. En effet, différents langages peuvent être utilisés pour écrire les moniteurs et

les tâches. Il convient donc de spécifier quel langage a été utilisé pour le cas décrit.

✓ Plates-formes supportées :
Description des plates-formes techniques sur lesquelles le programme peut être exécuté. Cette description comprend le type ainsi que la version du matériel et du système d'exploitation.

✓ Localisation
Spécification de l'endroit où réside la source du programme et de son installation. Les sources peuvent se trouver soit dans l'outil de gestion des sources du département ESM, soit sur une plate-forme appartenant à un autre département. Pour ces localisations, le tableau suivant est utilisé :

Machine	Nom	Chemin d'accès

Où : Machine — Nom de la ou des machines où se trouve la source et où doit être installé le programme.

Nom réel — Nom complet du programme avec ses éventuelles extensions.

Chemin d'accès — Chemin d'accès complet dans l'outil de gestion des sources ou sur la plate-forme où réside la source ou où le programme sera installé.

✓ Autorisations : autorisations, c'est-à-dire droits d'accès, nécessaires pour pouvoir exécuter le programme. Deux types de droit d'accès sont à spécifier :

- Les droits d'accès aux ressources. Ces droits sont décrits dans le tableau suivant :

Ressource	Autorisations	Machine

Où : Ressource — Type et nom de la ressource. Par exemple, un fichier ou un répertoire.

Autorisations — Type d'autorisation. Par exemple, accès en lecture, écriture, ...

Machine — Nom des machines sur lesquelles de tels droits doivent être définis.

- La spécification des groupes de sécurité auxquels l'utilisateur technique[85] doit appartenir pour l'exécution le programme (tâche, moniteur, ...) décrit.

Groupe	Type	Machine

Où : Groupe — Nom du groupe de sécurité.

[85] Tous les programmes des moniteurs et des tâches sont exécutés par les outils de supervision en utilisant un identifiant utilisateur particulier appelé utilisateur technique.

Type	Type du groupe : local ou domaine, par exemple.
Machine	Nom des machines sur lesquelles de tels droits doivent être définis.

Exécution

La section consacrée à l'exécution du programme va décrire l'environnement, les prérequis et les modes d'exécution du programme. Elle contient les deux sections suivantes :

- ✓ Prérequis
  Liste des prérequis nécessaires pour l'exécution du programme. Ceux-ci peuvent être des ressources disponibles, des logiciels, des variables systèmes, ...
- ✓ Mode d'exécution
  Description du mode d'exécution du programme. Un programme peut, par exemple, n'être exécuté qu'une seule fois, plusieurs fois mais avec une procédure de retour en arrière ou plusieurs fois sans aucune restriction. Il faut également spécifier s'il peut être exécuté à la demande, appelé par un autre programme ou moniteur, planifié ou appelé par une tâche. Enfin, il est nécessaire de signaler si le programme peut être utilisé sur des systèmes spéciaux tels que les systèmes en grappe ou dans des zones sécurisées.

Description technique

Cette partie décrit les paramètres, les entrées et sorties ainsi que les différentes actions exécutées par le programme.

- ✓ Paramètres :
  Description des paramètres obligatoires et/ou optionnels du programme. Ces paramètres peuvent également être positionnels ou passés au programme via des commutateurs[86].
- ✓ Entrées et Sorties :
  Noms et localisations des valeurs de retour, des fichiers, journaux électroniques utilisés par le programme.
- ✓ Actions :
  Description des différentes sous-routines du programme.

Procédure d'échec

Cette section décrit, sous forme de texte ou de pseudo-code, les procédures éventuelles à suivre en cas d'échec afin, par exemple, de remettre l'environnement dans l'état dans lequel il se trouvait avant son exécution.

4.4.5 Formulaire des pages d'aide

Ce formulaire est destiné à décrire les procédures que va devoir suivre la *Master Console* en réponse à une alarme pour résoudre le problème. Tant la forme que le contenu de ces pages doivent être validés par un représentant de la *Master Console*. L'utilisation d'un formulaire pour l'élaboration de ces pages est intéressante à plus d'un titre :

- ✓ Le contenu et la forme doivent, pour chaque nouvelle page proposée, faire l'objet d'une validation. L'utilisation d'un formulaire et donc d'un format standard de présentation de page validé par la *Master Console*, va réduire le

[86] Par exemple : –m "texte du message".

temps de validation des pages. En fait, la forme étant validée une fois pour toutes, seul le contenu devra encore faire l'objet d'une validation.

- ✓ L'utilisation d'un format unique de présentation de ces pages va permettre une lecture plus aisée des procédures de résolution de pannes. A terme, cela va mener à un meilleur temps de réaction aux pannes et donc à un accroissement d'efficacité de la part de la *Master Console*.

A noter que la rédaction de ces pages est de la responsabilité du client. En effet, il est évident qu'il est le plus à même de décrire les procédures à suivre afin de déterminer et de résoudre le problème associé à l'alarme.

Le formulaire devra permettre de déterminer la procédure à suivre en fonction :

1. Du moment de la journée, de la semaine, ... durant lequel le problème survient. En d'autres mots, le formulaire devra permettre de prendre en compte des notions de calendrier.
2. Du contenu de l'alarme. Dans la plupart des cas, le libellé d'une alarme est lié à un type de problème et non à une ressource, par exemple, une alarme peut signaler la perte d'une session. L'identification de la ressource incriminée dans ce cas, le nom de la connexion, est, quant à elle, renseignée dans les champs de l'alarme. Ainsi, il se peut que pour une même alarme, la procédure à suivre soit différente suivant la ressource impliquée. Le formulaire doit permettre de définir soit une procédure unique pour le type d'alarme, soit plusieurs procédures déterminées en fonction des données présentes dans l'alarme.

Le formulaire pour la rédaction de telles pages se présente comme suit :

Données de l'auteur

Cette section contient les références (nom et département) de l'auteur de la page ainsi que la date de sa dernière mise à jour. Ces informations sont principalement utilisées afin de savoir à qui adresser des demandes éventuelles de modification des procédures, suite notamment à un changement d'environnement.

Description de l'alarme

Cette partie décrit l'alarme concernée par cette page. La description contient les informations suivantes :

- ✓ Texte : Texte de l'alarme tel qu'il est visible sur le serveur d'alarmes. Cette information sert de lien entre le serveur d'alarmes et la page d'aide.
- ✓ Source : Indique la source de l'alarme. Il ne s'agit pas ici de donner le nom de machine d'où provient l'alarme mais plutôt le système à l'origine de l'alarme. Par exemple, un système d'exploitation, un système de gestion de base de données, un middleware, ...
- ✓ Description : Explication, dans les grandes lignes, de la signification technique de l'alarme, par exemple, s'il s'agit d'une alarme signalant la perte de connexion.
- ✓ Cause : Cause(s) probable(s) du problème ayant entraîné l'envoi de l'alarme.
- ✓ Impact : Influence du problème sur la disponibilité du SI, d'une plate-forme ou d'une middleware.

Appel d'urgence

Au cas où les procédures décrites ne fonctionneraient pas correctement ou ne permettraient pas de résoudre le problème, une liste de personnes à contacter doit être fournie. Pour ce faire, deux possibilités existent :

1. Le SI, l'application ou le middleware dont relève l'alarme, dispose d'un code applicatif dans le référentiel des applications. Ce référentiel, qui contient, entre autres, une liste ainsi qu'un ordre d'appel des personnes à contacter en cas de problème, est couramment utilisé par la *Master Console* pour les chaînes de programmes de nuit. Il suffit dans ce cas de spécifier dans la page le code applicatif du référentiel à consulter pour obtenir la liste d'appel. Cette méthode présente également l'avantage de garder les listes d'appel centralisées.
2. Dans le cas contraire, il convient de donner la liste des personnes à appeler en cas de problème sous la forme :

Nom/Prénom	Département	N° Interne	N° GSM/Privé

Où : Nom/Prénom : les coordonnées de la personne à contacter.

Département : le numéro et le nom du département.

N° Interne : le numéro de téléphone interne.

N° GSM/Privé : le numéro de téléphone privé et/ou de GSM professionnel.

Calendriers

Les procédures de réponse aux alarmes peuvent être différentes en fonction de l'heure ou du jour durant lesquels l'alarme est reçue. Par exemple, si un système est arrêté pour une prise de copie de sauvegarde chaque jour à une heure déterminée, il peut être demandé de ne pas tenir compte des éventuelles alarmes concernant ce système pendant ces heures. Par contre, si des alarmes arrivent en dehors de ces heures ou si elles ne sont pas clôturées à l'heure supposée de la fin de l'intervention, une autre procédure doit alors être exécutée. Ou plus simplement, les procédures à suivre pendant les jours fériés sont probablement différentes des procédures à utiliser les autres jours.

Ce genre de demande n'est pas rare, c'est pourquoi il est nécessaire de spécifier différentes périodes auxquelles seront associées des procédures particulières.

Ces calendriers se décrivent comme suit :

Calendriers		
		Procédure n

A chaque ligne du tableau correspond la définition d'une période calendrier. Chaque ligne est divisée en 3 colonnes contenant les informations suivantes :

- ✓ Le numéro d'ordre du calendrier.
- ✓ La spécification de la ou des périodes du calendrier.
- ✓ Un *hyperlien* permettant de se déplacer dans la page directement à l'endroit où est décrite la procédure à suivre pour la période concernée.

Si la procédure à suivre est identique quelle que soit la période de temps, le tableau ne contiendra qu'une seule ligne :

1	Réponse inconditionnelle 7 jours sur 7 et 24 heures sur 24	Procédure unique

Procédures

La partie consacrée aux procédures contient les descriptions des procédures à suivre à raison d'une description par période calendrier définie. Cette description se présente également sous forme de tableau :

1	...
Conditions	Procédure

La première ligne (indiquée par ...) reprend le numéro ainsi que la description de la ou des périodes du calendrier telles que spécifiées dans le tableau des calendriers.

Les lignes suivantes contiennent les descriptions des procédures à suivre avec les informations suivantes :

- ✓ Conditions Description d'une ou plusieurs conditions d'égalité ou d'inégalité que doivent respecter les valeurs d'un ou plusieurs champs de l'alarme pour que la procédure puisse être appliquée. Pour énoncer ces conditions, le formalisme suivant peut être utilisé :

  SI Nom_de_Champs CONDITION Valeur
  { [OPERATEUR LOGIQUE BINAIRE Nom_de_Champs CONDITION Valeur], ... }

- ✓ Procédure Description de la procédure à suivre pour résoudre le problème dans le cas où TOUTES les conditions mentionnées seraient respectées.

De même que pour les calendriers, si aucune condition n'est liée à l'alarme, c'est-à-dire s'il n'existe qu'une seule procédure quel que soit le contenu de l'alarme, le tableau ne contiendra qu'une seule ligne :

DANS TOUS LES CAS	Exécuter la procédure décrite ici ...

4.4.6 Conclusion

A la fin de cette phase, tous les développements relatifs à la supervision sont terminés et les tests peuvent commencer.

4.5 Phase IV : Tests et acceptation

4.5.1 Objectifs

Les objectifs de cette phase sont :

- ✓ Elaborer des plans de test de tous les développements (moniteurs, tâches, règles de corrélations...) issus de la phase précédente.
- ✓ Effectuer tous les tests.
- ✓ Valider la solution de supervision pour sa mise en production.

4.5.2 Plan de la phase

La phase IV se découpe comme suit :

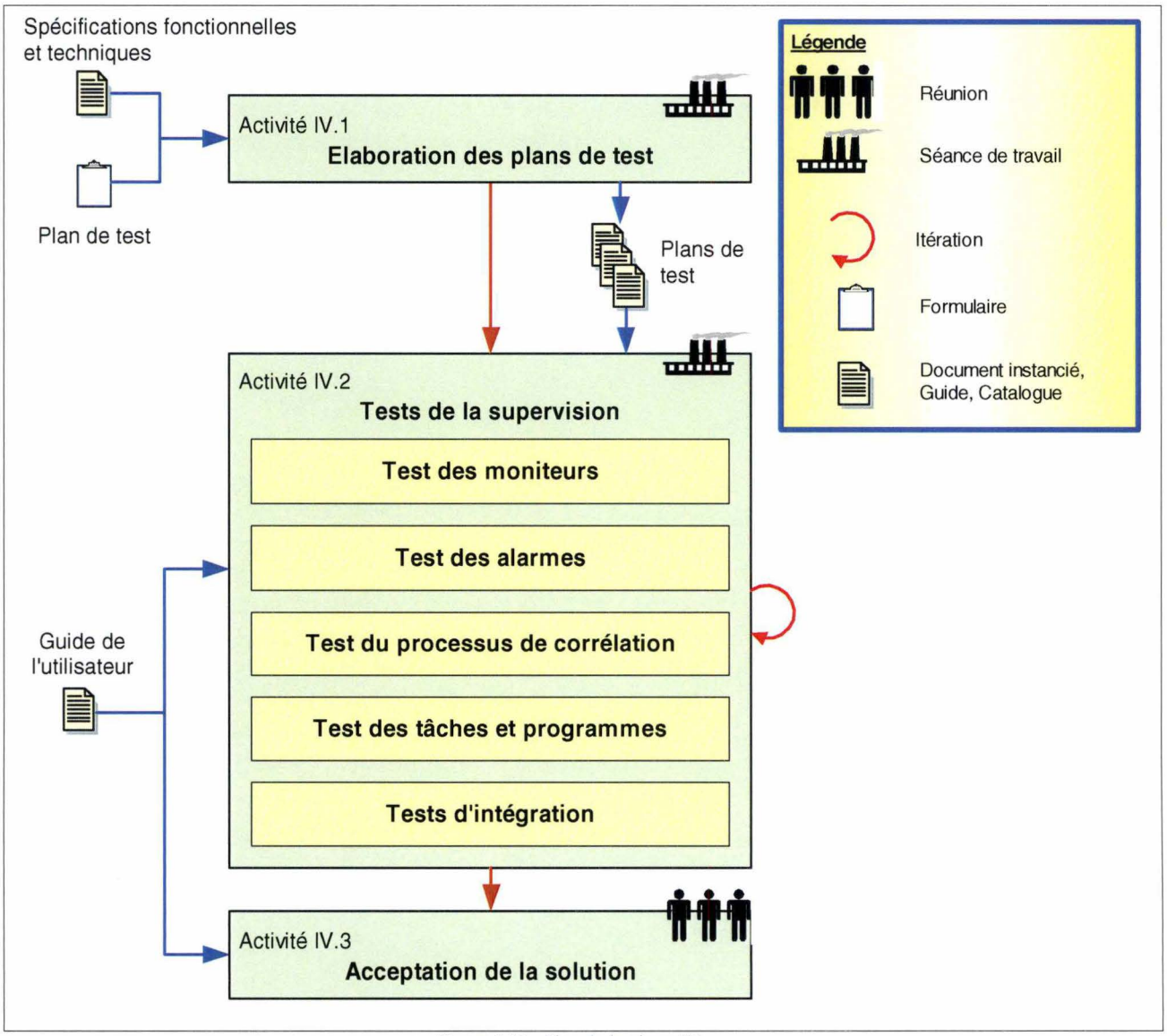


Figure 4-11 Plan de la phase IV

4.5.3 Activité IV.1 : Elaboration des plans de test

Tous les tests à effectuer peuvent être regroupés en quatre grands groupes :

- ✓ Les tests des moniteurs.
- ✓ Les tests des alarmes.
- ✓ Les tests du processus de corrélation.
- ✓ Les tests des tâches et des programmes.

Les quatre groupes de test sont à effectuer dans l'ordre dans lequel ils sont cités. Cet ordre est basé sur une méthode qui peut être assimilée à une méthode *Bottom-Up* qui part de la source de la supervision jusqu'au point central d'arrivée et de traitement des alarmes : le serveur d'alarmes. On calque en fait les tests sur la philosophie de la supervision décrite dans les notions de supervision[87]. Les tests vont donc s'effectuer par incrémentation, du plus près de la source jusqu'au serveur d'alarmes.

L'objectif de cette activité sera d'établir un plan de test particulier pour chacune de ces catégories et de test d'intégration pour la chaîne complète de supervision, du moniteur jusqu'à l'exécution d'une tâche automatique éventuelle sans oublier d'y inclure le test des règles de corrélation. Ces tests d'intégration serviront également de tests de validation de la solution de supervision. Si les quatre groupes de tests sont à effectuer par la cellule *Availability*, les tests d'intégration sont, quant à eux, à effectuer en collaboration avec le client.

Pour la spécification de tous ces jeux de test, un formulaire classique tiré d'une méthodologie de gestion de projet comme Merise peut être utilisé.

4.5.4 Activité IV.2 : Tests de la supervision

Par catégorie, les tests vont se dérouler comme suit :

Test des moniteurs

Les tests des moniteurs sont assez simples à mener. En effet, ces programmes se limitent à contrôler l'état d'une ressource. Celle-ci peut être disponible, indisponible ou dans un état intermédiaire. Les tests vont se limiter dans un premier temps à placer la ressource dans chacun de ces états et de voir si celui-ci détecte bien chacun d'eux. Dans un second temps, la ressource sera placée dans des états sortant de la spécification du moniteur, si bien sûr de tels états existent, afin de s'assurer que celui-ci ne réagit pas. Le choix du type de test à mener dans ce cas sera naturellement de type de test *Black Box*. En fait, les moniteurs retournent soit une valeur binaire ("est disponible", "est indisponible") soit une valeur de taille ou de capacité. Le choix d'un type de test fonctionnel, tels que les tests *Black Box*, basé sur les spécifications des programmes, permet de limiter les jeux de test au minimum.

Enfin, afin de faciliter les tests, la diminution des intervalles et des seuils des moniteurs est acceptée afin de diminuer le temps d'attente entre deux tests de ressource ou de permettre la génération d'alarmes pour des situations difficiles à provoquer (occupation d'un processeur, par exemple).

Test des alarmes

Les tests des alarmes consistent pour chaque état prévu de la ressource à s'assurer :

- ✓ Que la bonne alarme est générée en réponse au résultat d'un moniteur.
- ✓ Que tous les champs de cette alarme sont présents et correctement remplis.

Pour ce faire, on placera une nouvelle fois la ressource dans chaque état attendu afin de vérifier que chaque état provoque bien la génération de la bonne alarme et que les

[87] Voir Notions de supervision page 19.

champs attendus pour ces alarmes contiennent les bonnes valeurs. Pour vérifier la génération correcte des alarmes, on se basera sur les spécifications fonctionnelles des supervisions et, pour le contenu des champs des alarmes, sur le tableau de spécification technique des alarmes[88] pour le contenu de celles-ci. Enfin, il ne faudra pas oublier de vérifier que les champs sont bien remplis aux endroits (*Source* ou *T/EC*) tels que spécifiés dans ce dernier document.

Test du processus de corrélation

Les tests du processus de corrélation ont comme objectifs de vérifier que :

- ✓ Les corrélations se déroulent telles que spécifiées dans les réseaux des relations.
- ✓ Les alarmes identiques sont correctement détectées et le compteur d'alarmes correctement incrémenté.
- ✓ Les actions automatiques telles que la génération d'un ticket d'incident ou le redémarrage automatique sont bien exécutées et, pour les tickets, que le *SCIM* et le destinataire sont corrects.

Pour mener à bien tous ces tests, on placera encore une fois toutes les ressources supervisées dans les états attendus afin de provoquer la génération de toutes les alarmes et vérifier dans quelle mesure les règles de corrélation sont correctes et les tâches automatiques exécutées.

Test des tâches et programmes

Les tâches sont des programmes de très petite taille (une dizaine de ligne de code en moyenne) qui sont utilisées afin d'intervenir, à partir d'un point central, sur les ressources. Chacune d'elles a une fonction unique, par exemple, une tâche sera utilisée pour arrêter une ressource, une autre pour la redémarrer. Les tests de ces tâches sont donc très simples puisqu'ils vont se limiter à vérifier que la tâche a bien arrêté ou démarré la ressource tel que spécifié dans le dossier fonctionnel.

Test d'intégration

Les tests d'intégration sont en fait des tests d'acceptation. Ils sont menés par la cellule *Availability* et par le client afin de vérifier la validité de l'ensemble de la solution de supervision. Ils consistent en un groupement des quatre autres catégories de tests.

4.5.5 Activité IV.3 : Acceptation de la solution

L'acceptation de la solution de supervision est le résultat des tests d'intégration. Si ceux-ci sont positifs, le client donnera le feu vert pour la phase suivante du projet : le déploiement.

4.5.6 Conclusion

Après cette phase, tous les tests ont été effectués et la solution a été acceptée par le client. Rien n'empêche plus le déploiement de la solution de supervision.

---

[88] Voir Formulaire de spécification technique page 108.

4.6 Phase V : Déploiement et suivi de la solution

4.6.1 Objectifs

Les objectifs de la phase de déploiement et de suivi de la solution de supervision sont les suivants :

- ✓ Assurer la formation de la *Master Console*.
- ✓ Etablir un plan de déploiement pour la supervision et mener à bien ce déploiement.
- ✓ Assurer un suivi du déploiement de la supervision et de la qualité de la solution.

4.6.2 Plan de la phase

Les différentes activités de la phase sont organisées comme suit :

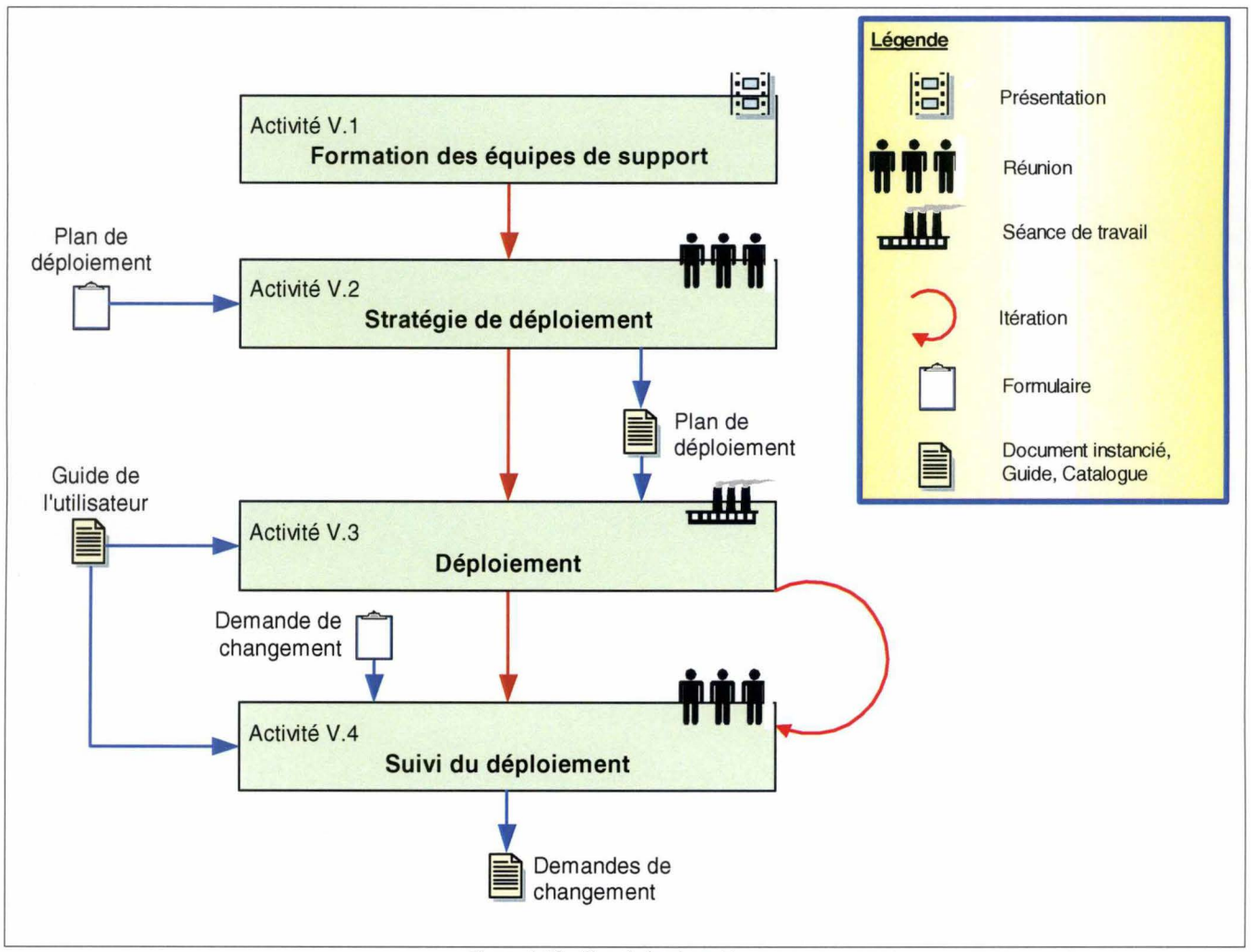


Figure 4-12 Plan de la phase V

4.6.3 Activité V.1 : Formation des équipes de support

Dans l'optique d'un support de qualité, la formation de la *Master Console* est primordiale. Celle-ci, pour des raisons évidentes, doit absolument être terminée avant le déploiement de la solution.

Cette activité de formation, de type présentation, s'articulera autour de cinq axes :

Description du SI

Cette formation a comme objectif de faire connaître le SI de manière la plus complète possible à la *Master Console*. Elle comprend une description fonctionnelle du SI, de sa topologie, des composants qui le composent ainsi que des connexions entre ces composants. Cette formation peut en fait être construite autour des documents produits lors de la phase I : les descriptions logique et technique du SI[89].

Description de la supervision

La description de la supervision consiste à informer les équipes de support sur ce qui est et comment est supervisé le SI. On indiquera également les composants et ressources dont la responsabilité de surveillance et de résolution de pannes incombera à ces équipes de support.

Description des alarmes

Il s'agit ici de présenter les principales alarmes susceptibles d'être générées par la supervision du SI et d'indiquer les champs les plus importants[90] de ces alarmes. On décrira également, dans les grandes lignes, l'origine ainsi que la cause probable de la panne signalée par l'alarme. Une description des corrélations fait également partie de cette formation.

Description des tâches

Il s'agit de décrire la fonction, les entrées et les sorties de toutes les tâches écrites pour la supervision.

Description des procédures

Enfin, il va falloir expliquer dans le détail toutes les procédures de résolution des pannes décrites dans les pages d'aide fournies à la *Master Console*. C'est la partie la plus importante de la formation car, pour obtenir un support de qualité, il ne peut subsister aucun doute quant à la manière de comprendre ou d'exécuter ces procédures.

---

89 Voir Activité I.3 : Description du système d'information page 73.

90 Champs contenant de l'information utile pour l'identification de la ressource ou du composant défectueux ou nécessaire pour l'exécution d'une procédure à exécuter pour résoudre le problème.

4.6.4 Activité V.2 : Stratégie de déploiement

4.6.4.1 Objectif

Cette activité de type réunion consiste à élaborer la stratégie de déploiement de la solution de supervision et à planifier ce déploiement. Celui-ci comprend dans la plupart des cas quatre parties :

- ✓ L'installation de l'architecture de supervision propre au SI.
- ✓ L'installation des règles de corrélation.
- ✓ L'installation des tâches et des programmes.
- ✓ Le déploiement de la supervision, autrement dit l'installation des moniteurs sur tous les composants du SI.

Les trois premières parties peuvent être installées en une fois sans problème même sans l'accord du client puisqu'elles sont propres aux outils de supervision. Par contre, pour le déploiement de la supervision, c'est-à-dire des moniteurs, sur les différents composants la cellule *Availability* doit obtenir l'aval du client tant sur la planification de ce déploiement que sur la manière dont il va se dérouler.

4.6.4.2 Types de déploiement

Pour l'installation des moniteurs, trois modes seront proposés au client :

- ✓ Déploiement en un seul coup.
  Ce type de déploiement consiste à installer tous les moniteurs, les règles de corrélation et les tâches en une seule fois. Ce genre de déploiement est parfois dangereux si le nombre de composants du SI est important. Ainsi, l'installation en une seule fois de la supervision sur ces composants va également provoquer l'activation en une seule fois de tous les moniteurs définis pour ces composants. Cette activation en masse est susceptible de provoquer une avalanche importante d'alarmes qui risque à terme d'inonder le serveur d'alarmes. Si ce type de déploiement est à déconseiller pour les supervisions importantes pour lesquelles on préférera un mode de déploiement plus progressif, il est cependant tout à fait acceptable pour les supervisions de petite taille.
- ✓ Déploiement progressif.
  Il s'agit du déploiement à privilégier pour les supervisions de taille importante. Il consiste à regrouper les ressources au sein de groupes de déploiement. Chaque groupe sera déployé l'un après l'autre avec, entre chaque déploiement, une période d'attente de stabilisation de la supervision. Cette période d'attente est en fait utilisée pour permettre à tous les moniteurs nouvellement installés de générer les éventuelles alarmes et à la supervision de prendre sa "vitesse de croisière". Généralement, le premier groupe contiendra un échantillon de chaque type de composant afin d'avoir un embryon de la supervision couvrant tous les types de composant.
- ✓ Déploiement par type de composant.
  L'idée est de déployer la supervision par type de composant. Par exemple, déployer la supervision par type de système d'exploitation, puis par type de middleware et enfin par application. Ce mode est intéressant pour les supervisions importantes car il est progressif ainsi que pour les SI hétérogènes car il permet d'obtenir des supervisions de type de composant en une seule fois.

Quel que soit le déploiement choisi, il faudra de toute manière installer en premier lieu et en une fois les règles de corrélation et les tâches. En effet, celles-ci s'installent en un point

central. Il n'y a aucune raison de diviser leur installation. Il en est de même pour les pages *URL* d'aide à la résolution de problèmes. TOUTES ces pages d'aide doivent être fournies en une seule fois, et ceci afin d'être sûr d'en disposer avant tout déploiement, avant même l'installation des règles de corrélation et des tâches.

4.6.4.3 Formulaire de description du déploiement

Pour décrire les différentes étapes du déploiement, un formulaire spécifique est utilisé. Celui-ci comprend trois parties :

- ✓ Déploiement des pages *HTML.*

  Ce déploiement est décrit dans le tableau suivant :

Déploiement des pages HTML		
Etapes	Date début	Date fin
Mise à disposition des pages		
Installation des pages		

Le déploiement des pages *HTML* se déroule en deux étapes :

- ✓ Une mise à la disposition de la cellule *Availability* par le client des pages *HTML*.
- ✓ L'installation de ces pages dans le système de documentation.

Pour chaque étape, sont spécifiées une date de début et une date de fin. Par exemple pour la première étape, ces dates indiquent les dates au plus tôt et au plus tard auxquelles sont attendues ces pages.

- ✓ Déploiement de l'environnement.

  Le déploiement de l'environnement consiste à installer l'architecture de supervision, les règles de corrélation ainsi que les tâches liées à la supervision. Toutes ces ressources doivent être installées avant le déploiement des moniteurs. Le tableau suivante est utilisé pour la description du déploiement.

Déploiement de l'environnement		
Etapes	Date début	Date fin
Installation de l'architecture		
Installation des règles de corrélation		
Installation de tâches		

- ✓ Déploiement de la supervision

  Le déploiement de la supervision va décrire le type ainsi que les échéances du déploiement des moniteurs. Ce déploiement est décrit dans le tableau suivant :

Déploiement de la supervision		
Type de déploiement		
Groupes	Date début	Date fin

Où : Type de déploiement est le type de déploiement choisi.

Groupes décrit les groupes de composants créés pour le déploiement. Si un déploiement en un seul coup est préféré, un seul groupe "GENERAL" sera spécifié. Pour chaque groupe, on spécifiera une date de début de déploiement et une date de fin pour laquelle le déploiement de la supervision pour le groupe doit être terminé.

4.6.4.4 Participants

Le choix d'un mode de déploiement doit être fait par la cellule *Availability* en accord avec le client. C'est pourquoi, en plus de la présence d'un représentant de la cellule, celle du chef de projet ou du correspondant technique s'avère nécessaire.

4.6.5 Activité V.3 : Déploiement

Le déploiement est une activité de type séance de travail qui consiste à installer la supervision sur les différents composants du SI suivant les plans de déploiement établis lors de l'activité précédente. Ce déploiement est exclusivement exécuté par la cellule *Availability*.

4.6.6 Activité V.4 : Suivi de la supervision

Le suivi de la supervision est une activité de type réunion ayant comme objectif d'assurer un suivi du déploiement et de la qualité de la supervision.

Le suivi du déploiement sont des réunions entre la cellule *Availability* et le chef de projet du SI durant lesquelles sera contrôlé le bon déroulement de l'installation des moniteurs sur les différents composants du SI. Il s'agit de voir si ce déploiement s'est déroulé dans de bonnes conditions et dans les délais prévus dans les plans de déploiement. Si un déploiement progressif ou par type de composant est choisi, une réunion de suivi sera organisée après chaque déploiement d'un groupe.

Les réunions de suivi de qualité de la supervision sont organisées après l'installation complète de la supervision à des intervalles réguliers choisis par la cellule *Availability* en accord avec le chef de projet du SI. Elles ont comme objectif de passer en revue les principaux problèmes de fonctionnement qu'a connu le SI depuis la mise en place de la supervision. Pour chaque problème, la solution de supervision sera évaluée pour voir dans quelle mesure la supervision fut efficace pour la détection et la résolution du problème. Cette évaluation peut mener à l'introduction d'une demande de changement dont la charge de travail sera évaluée afin de déterminer si ce changement peut être appliqué directement ou être inclus dans une phase ultérieure du projet. Pour ces demandes de changement, le formulaire utilisé dans l'étude de faisabilité[91] peut être repis.

4.6.7 Conclusion

Mis à part les demandes de changement produites lors des réunions de suivi de qualité qui seront implémentées dans une phase suivante du projet ou feront l'objet d'un projet séparé, la solution de supervision ou tout au moins l'une de ses phases est totalement déployée et opérationnelle.

---

[91] Voir Formulaire de demande de changement page 101.

4.7 Conclusions

Des objectifs que nous nous étions fixés pour la méthodologie[92], seuls les trois derniers points restaient à atteindre. En effet, les deux premiers objectifs avaient déjà été atteints par la mise en place d'une architecture, de règles et de profils structurés de supervision[93].

Le troisième objectif[94] a clairement été atteint par l'utilisation d'un système de documentation structuré[95] et, tout au long des cinq phases de la méthodologie, de formulaires standardisés.

Pour le quatrième objectif[96], la minimisation des développements est rendue possible grâce à l'utilisation d'outils de supervision existants découverts lors de l'enquête technique[97] tandis que la réutilisation de l'existant trouvera sa solution par l'utilisation du catalogue des supervisions existantes. En effet, celles-ci sont installées obligatoirement sur tout composant couvert par une telle supervision ce qui permet un réemploi optimum de l'existant.

Enfin, le cinquième et dernier objectif[98] a été atteint par les activités de planification de la supervision de l'étude de faisabilité de la phase II, les réunions de synchronisation de la phase III et enfin les réunions de suivi de déploiement de la phase V de la méthodologie.

Reste maintenant à voir comment la méthodologie peut être appliquée dans les principaux types de projets auxquels est habituellement confrontée la cellule *Availability*.

---

92 Voir les conclusions du chapitre Audit page 26.
93 Voir Choix Stratégiques page 41.
94 Définir des standards pour la spécification et la documentation du projet afin d'améliorer le dialogue avec l'utilisateur et de guider celui-ci d'un bout à l'autre du projet.
95 Voir Les documents page 63.
96 Minimiser les développements liés au projet de supervision et permettre un réemploi maximum des supervisions existantes.
97 Voir Activité II.3 : Enquête technique page 104.
98 Evaluer une planification de développement et aider à respecter les délais.

5 Mise en pratique de la méthodologie

5.1 Introduction

La cellule *Availability* est confrontée à un nombre important de demandes pour des projets de supervision qui peuvent être classées en trois catégories :

- ✓ Supervision de plates-formes techniques, middleware, ...
- ✓ Supervision d'une application existante.
- ✓ Supervision d'une nouvelle application en cours d'analyse ou de développement.

Pour chacun de ces trois types de demande, il peut être intéressant de voir dans quelle mesure la méthodologie doit être appliquée, de voir quelle(s) phase(s) et/ou activité(s) peuvent ou doivent être omises ou imposées.

5.2 Mise en pratique

5.2.1 Supervisions techniques

Les projets de supervisions techniques consistent à mettre en œuvre des supervisions pour des plates-formes techniques, des systèmes d'exploitation ou des middleware.

Ces projets se différencient des autres projets par le fait que :

- ✓ Ceux-ci concernent un type unique de composant. En effet, les composants faisant partie de tels projets sont tous identiques. Ainsi il sera demandé d'élaborer, par exemple, une supervision pour un système d'exploitation X, un middleware Y présent sur un système d'exploitation X, ...
- ✓ Les composants à superviser sont nombreux. En effet, de tels projets sont destinés à superviser TOUS les composants d'une même famille au niveau entreprise. Par exemple, dans le cas de FORTIS, superviser tous les serveurs *NT* équivaut à superviser entre quatre et cinq milles plates-formes.

Pour ces raisons, de tels projets vont amener quelques adaptations ou imposer des choix dans la méthodologie.

Phase I : Interview du client

Les deux premières activités de cette phase sont des présentations d'informations à destination du client. La présentation de sensibilisation à la problématique de la supervision peut sans conteste être supprimée. En effet, dans la majorité des cas, le client fait déjà partie du niveau I.2 de support et est donc informé du fonctionnement de ce support. La présentation de la méthodologie doit, quant à elle, être maintenue puisqu'elle sera également utilisée pour de tels projets.

Les activités I.3 et I.4 dédiées à la collecte des descriptions logique et physique du SI peuvent également être omises. En fait, on parle ici d'un parc composé d'un grand nombre de plates-formes techniques de même type n'ayant dans la plupart des cas aucun lien fonctionnel ou physique entre elles. De telles descriptions n'ont ici aucun sens et aucun intérêt. C'est pourquoi ces activités peuvent être supprimées.

L'activité de spécification fonctionnelle de la supervision, quant à elle, est bien sûr nécessaire et doit être maintenue. Comme nous l'avons vu dans les choix stratégiques, une procédure de définition de profils de supervision[99] a été mise au point afin de diminuer leur nombre. Pour les supervisions techniques dans lesquelles le nombre de plate-forme est très important et où le risque de multiplier le nombre de profils de supervision est grand, il est vivement conseillé d'utiliser cette procédure. En fonction du nombre de profils dégagés, il sera alors nécessaire de décider si la suite du projet doit être menée en une seule fois pour tous les groupes de composants définis ou en plusieurs étapes, groupe par groupe.

Phase II : Etude de faisabilité

L'étude de faisabilité ne va pas connaître de grands changements. Seule l'activité II.1 de restructuration des spécifications fonctionnelles[100] peut éventuellement être supprimée. En effet, de par l'homogénéité des composants à surveiller, les ressources à superviser sont très ciblées, par exemple, des services propres à un système d'exploitation ou un middleware. Ainsi, le risque est minime de rencontrer des demandes de supervision pour des ressources appartenant à une couche autre que la couche à laquelle appartient la plate-forme technique.

---

[99] Voir Profils de supervision page 52.
[100] Voir Activité II.1 : Restructuration des spécifications fonctionnelles page 100.

Pour l'activité II.5 de planification, si une découpe du projet est tout de même souhaitée en raison d'une charge de travail importante, une découpe du projet sur base des groupes dégagés lors des spécifications fonctionnelles peut de nouveau être proposée.

Phase III : Développement de la supervision

La phase III de la méthodologie peut être suivie sans modification. Il faut cependant noter que les moniteurs destinés à un grand nombre de plates-formes. Il est naturel de remplacer dans les documents de description technique des programmes[101], le nom de machine par l'identification d'un groupe défini lors de la phase I.

Phase IV : Tests et acceptation

Cette phase peut également être suivie dans son intégralité sans nécessiter de modifications.

Phase V : Déploiement et suivi de la solution

Si la phase V peut également être suivie dans son entièreté, le choix du mode de déploiement s'impose de lui-même pour ce genre de projet. En effet, le nombre de plates-formes concernées peut être très élevé. Il est dès lors suicidaire d'opter pour un déploiement en un seul coup car le risque est réel de noyer le serveur d'alarmes lors de l'activation des moniteurs après leur installation. Il faudra opter pour un déploiement progressif ou par type de composant car la notion de type de composant peut être associée à celle de groupe défini lors de l'élaboration des spécifications fonctionnelles.

[101] Voir Formulaire de documentation technique des programmes page 114.

5.2.2 Nouvelles applications

Pour ces projets, la méthodologie peut être utilisée telle quelle. Il convient cependant de voir comment les différentes phases et activités de la méthodologie peuvent se greffer dans le cycle d'un projet de développement. Pour illustrer les propos, nous allons prendre comme exemple un cycle de développement largement utilisé au sein de l'entreprise : le cycle de développement en V. L'imbrication de la méthodologie dans ce type de projet est présentée dans la figure suivante :

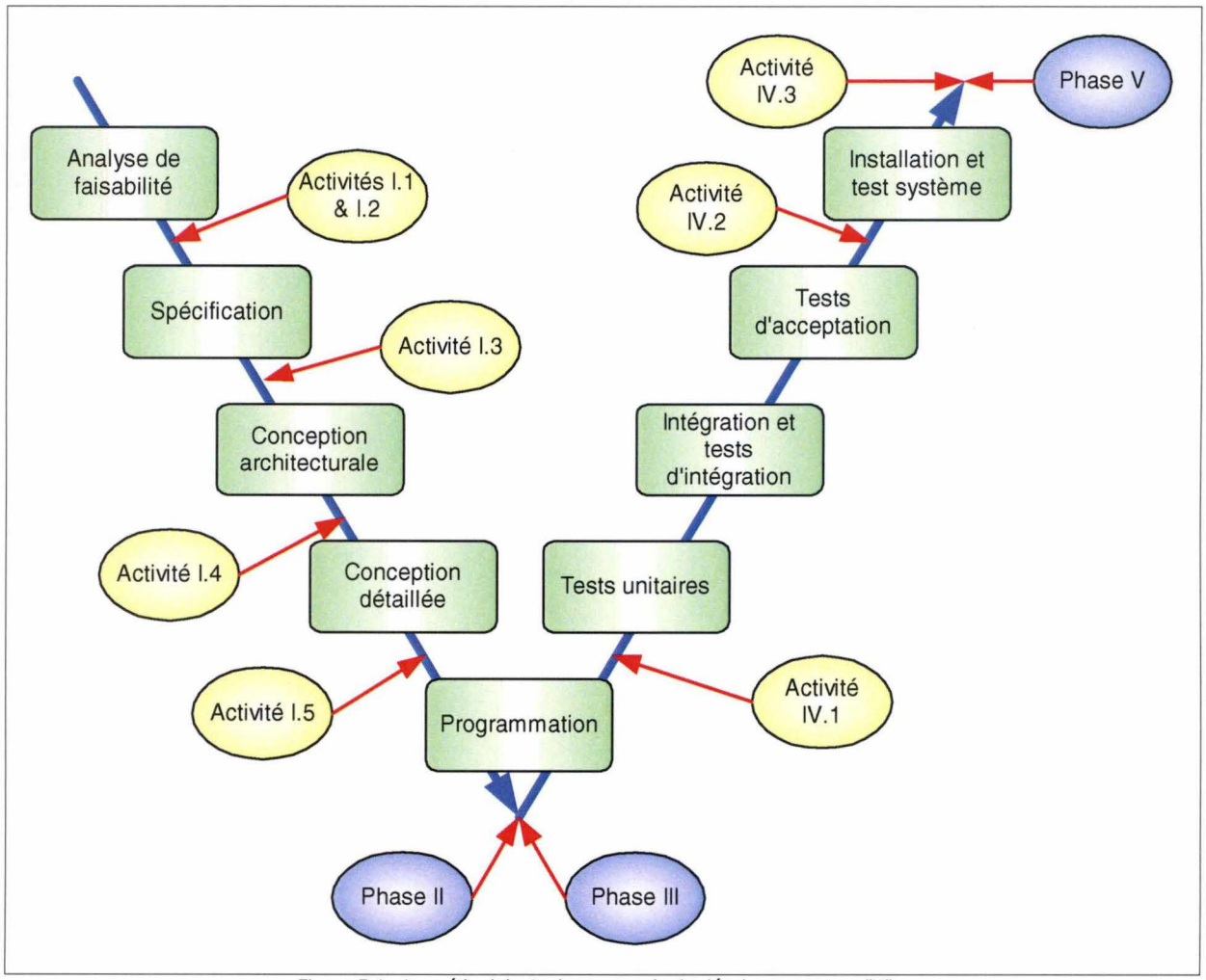


Figure 5-1 La méthodologie dans un cycle de développement en "V"

Les activités de la phase I se répartissent entre la fin de l'étude de faisabilité du projet et le début de la phase de programmation. En effet, un projet n'est réellement lancé que lorsque son étude de faisabilité est terminée et qu'un avis favorable est donné quant à la poursuite du projet qui sera également le signal de lancement du projet de supervision. Les activités se répartissent dans le cycle comme suit :

- ✓ Les activités I.1 et I.2 seront organisées dès le début du cycle de développement du projet, autrement dit, directement après l'étude de faisabilité. En effet, l'activité I.1 doit être menée le plus tôt possible dans ce cycle car, durant cette activité peuvent être prodigués des conseils quant au design et à la programmation de l'application qui peuvent faciliter sa supervision.
- ✓ Une fois fixées l'architecture et l'organisation du projet, ces informations peuvent être traduites dans les documents propres à la méthodologie. C'est pourquoi l'activité I.3 peut dès lors être organisée après l'étape de spécification du projet.

- ✓ La description physique dressée lors de l'activité I.4 peut être menée à bien dès la fin de la conception architecturale de l'application.
- ✓ Enfin, les spécifications fonctionnelles de la supervision collectée lors de l'activité I.5 peuvent être complétées en même temps que les spécifications de l'application.

Les phases II et III de la méthodologie, phases d'étude de faisabilité et de programmation, peuvent être menées tout au long de l'étape de programmation de l'application.

La phase IV sera menée juste après l'étape de programmation et ce, jusqu'à la fin du projet. Les activités de cette phase se répartissent comme suit :

- ✓ L'élaboration des plans de tests, l'activité IV.1, peut être menée dès la fin de la programmation.
- ✓ L'activité IV.2 regroupant les tests de la supervision doit être menée lorsque l'application est stable et n'est donc plus sujette à modification. Le moment idéal pour effectuer ces tests se situe juste avant le déploiement de l'application.
- ✓ Enfin, l'acceptation de la supervision, activité IV.3, peut avoir lieu avant ou, pour ne pas retarder le déploiement de l'application, après le déploiement de l'application.

La phase V de déploiement de la supervision se déroulera après le déploiement de l'application afin de permettre à cette dernière de se stabiliser.

L'idée principale à retenir pour de tels projets, est que le projet de supervision ne peut avoir une influence significative sur le projet à superviser tant sur le plan de design que sur la planification. La supervision ne peut être la source de modifications d'architecture importantes ou de retards dans la planification du projet. La difficulté sera de calquer la planification du projet de supervision sur celle de l'application malgré les éventuelles difficultés causées par l'architecture de cette dernière.

5.2.3 Applications existantes

C'est pour de tels projets que l'utilisation de la méthodologie peut être utilisée dans sa totalité avec un maximum de souplesse. La seule restriction sera une restriction de planification. En effet, dans la majorité des cas, la supervision sera à installer sur des plates-formes soumises à une gestion planifiée des versions. Sur de telles plates-formes, les mises à jour et les installations de programmes sont planifiées à l'avance à raison de quatre par an. Tout programme installé sur ces plates-formes doit faire l'objet d'une validation de la part de l'équipe technique responsable de cette plate-forme. Il faudra veiller à calquer la planification du projet de supervision sur celui de la gestion des versions de telle manière que les tests et les déploiements de ces deux projets coïncident.

Conclusions

En conclusion à cette étude, nous aimerions apporter quelques réflexions, d'une part, sur les méthodologies du marché qu'il nous a été donné d'étudier et, d'autre part, sur la méthodologie que nous avons conçue.

Comme nous l'avons vu, il existe un certain nombre de méthodologies qui abordent la problématique de la supervision. Même si nous n'avons repris dans cette étude que les méthodologies qui nous paraissaient les plus à mêmes d'apporter des solutions aux problèmes de la cellule *Availability* ou celles dont certains concepts pouvaient être repris dans notre méthodologie, nous avons remarqué une certaine constante parmi toutes ces méthodologies : elles sont assez récentes et ont été conçues par des constructeurs informatiques. Elles sont de type "propriétaires". Cette apparition dans un premier temps d'outils de supervision et dans un second temps des méthodologies, tend à prouver, d'une part, que les responsables d'entreprises ou de départements informatiques prennent de plus en plus conscience qu'il ne leur est plus permis de mettre en place des systèmes d'information sans une possibilité de contrôle en temps réel de leur fonctionnement et, d'autre part, que le métier de supervision est devenu un métier à part entière avec ses outils et ses méthodologies.

Enfin, nous regrettons de ne pas avoir eu la possibilité d'étudier plus en détail toutes ces méthodologies. En effet, étudier les 34 livres d'ITIL était impossible dans un délai raisonnable. Quant aux méthodologies d'IBM, le fait qu'elles soient toutes de type "propriétaire" a limité grandement les ressources librement accessibles qui auraient permis une meilleure étude. Pourtant, au vu du peu qu'il nous a été donné de consulter, beaucoup de composants de ces méthodologies, notamment les formulaires, auraient sans doute pu être réutilisables dans notre méthodologie, ce qui nous aurait probablement évité de réinventer la roue.

Pour ce qui est de la méthodologie que nous avons construite, nous admettons aisément qu'elle peut paraître lourde à l'emploi. Nous justifions cette lourdeur par le fait que nous avons dû répondre aux nombreux problèmes de la cellule *Availability* dont le plus important concernait la gestion de projet et notamment la collecte des informations. C'est pourquoi la méthodologie contient énormément de formulaires. Nous pensons également que l'utilisation systématique des formulaires tout au long du projet est le meilleur moyen, d'une part, d'obtenir toute l'information nécessaire et, d'autre part, de jalonner tout le parcours du projet de supervision et d'en accélérer ainsi les étapes.

Dans le même ordre d'idée, nous remarquons qu'une bonne partie de la phase d'interview du client est consacrée à la collecte d'informations destinées à dresser une carte d'identité du système d'information. Nos tentatives de construire la méthodologie en supprimant ou réduisant au maximum les activités de description logique et technique du SI (activités I.3 et I.4) pour alléger quelque peu la méthodologie n'ont pas été très heureuses car de nombreuses informations nécessaires manquaient alors. Bien que les informations demandées lors de cette activité soient destinées au projet de supervision, nous pensons tout de même qu'il n'est pas vraiment de la responsabilité du département ESM de collecter ce genre d'information. En effet, comme il en a été fait mention, il existe un référentiel des applications contenant déjà un certain nombre d'informations sur ces applications; il serait, nous semble-t-il, plus profitable que les informations collectées par la cellule *Availability* le soit plutôt dans le cadre de ce référentiel. Cela permettrait, d'une part, de centraliser toutes les informations disponibles sur les applications et de rendre accessibles ces informations à un plus grand nombre de personnes et, d'autre part, de décharger la cellule Availability d'une partie non négligeable de travail.

Enfin, une amélioration ou en tout cas une suite logique de la méthodologie serait de mettre en place un système d'information basé notamment sur un système de *Workflow* et une base de

donnée qui permettrait au client d'entrer directement les informations dans le système et d'éviter ainsi la manipulation de documents papier. Ce système permettrait d'utiliser les informations stockées dans la base de données afin, par exemple, d'automatiser certains développements ou de permettre l'étude d'impact d'un changement d'une alarme d'un composant sur les vues dans lesquelles il est présent.

Nous constatons également dans la pratique que si les chefs de projet prennent bien en compte les problématiques de copies de sauvegarde et d'archivage des données ainsi que de sécurité de leur système d'information, ils oublient fréquemment de prendre en compte la problématique de surveillance. Ce problème pourrait, nous semble-t-il, trouver sa solution dans l'instauration au niveau de l'organisation du département informatique d'un guichet unique. Celui-ci prendrait en charge pour tout chef de projet de SI, durant tout le cycle de développement, toutes les problématiques citées précédemment ainsi que la problématique de surveillance.

Pour terminer, nous admettons aisément que la méthodologie a été conçue dans un cadre de travail particulier et qu'elle est, de ce fait, étroitement liée à l'organisation tant des départements techniques que de support de la société pour laquelle elle a été conçue. Cependant, nous persistons à croire qu'elle pourrait être, moyennant quelques adaptations, appliquée dans d'autres organisations.

Bibliographies

CCTA, *The IT Infrastructure Library. An Introduction*, 5e édition, 1995.

COOK C., DARMAWAN B., FOSTER M., GILLARDO S., GUCER V., KONG D., KUMAR D., MANOEL E., PLASSMAN F., REYNOLDS R., SUGATANI K., YIU S., *An Introduction to Tivoli Enterprise*, IBM International Technical Support Organization, Octobre 1999.

FREAN P., BATTIS D., RICARDO BURGHI J., FELIGA A., GENERES T., HENDRY H., LANGBALLE E., *Designing Tivoli Solutions for End-to-End Systems and Service Management*, pages 19-34, IBM International Technical Support Organization, 1999.

TANENBAUM A., *Réseaux*, InterEditions, page 639, 3e édition, 1997.

THOENEN D., *Effective Design for Distributed Event correlation*, présentation pour Planet Tivoli 2001, Vienne.

TIVOLI Inc., *Tivoli Enterprise Concepts, Architecture, and services*, dossier d'introduction, 2000.

TIVOLI Inc., *Tivoli Enterprise Console User's Guide*, Documentation technique Tivoli, 1999.

Sites Internet

http://www.itil.co.uk.org Site officiel de ITIL.

http://www.redbooks.ibm.com Site de littérature technique IBM.

http://www.tivoli.com Site officiel de Tivoli Systems Inc.